**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**
**(Attorney Docket № 15815US02)**

In the Application of:

Peter Besen, et al.

Serial № 10/726,814

Filed: December 3, 2003

For: PROCESSING HIGH DEFINITION VIDEO DATA

Examiner: ROBERTS, JESSICA M

Group Art Unit: 2621

Confirmation № 4115

Customer № 23446

***Electronically filed on 24-NOV-2008***

**APPEAL BRIEF**

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is an appeal from an Office Action dated May 28, 2008 ("Final Office Action"), in which claims 1-15 were finally rejected. The Applicant respectfully requests that the Board of Patent Appeals and Interferences ("Board") reverses the final rejection of claims 1-15 of the present application. The Applicant notes that this Appeal Brief is timely filed within the two-month period for reply that ends on **November 24, 2008** (the Office date of receipt of the Notice of Appeal being September 24, 2008).

## REAL PARTY IN INTEREST
## (37 C.F.R. § 41.37(c)(1)(i))

Broadcom Advanced Compression Group, LLC, a limited liability company organized under the provisions and subject to the requirements of the Delaware Limited Liability Company Act, and having a place of business at 200 Brickstone Square, Suite 401, Andover, Massachusetts 01810, has acquired the entire right, title and interest in and to the invention, the application, and any and all patents to be obtained therefor, as set forth in the Assignment recorded at Reel 015111, Frame 0032 in the PTO Assignment Search room.

## RELATED APPEALS AND INTERFERENCES
## (37 C.F.R. § 41.37(c)(1)(ii))

The Appellant is unaware of any related appeals or interferences.

## STATUS OF THE CLAIMS
## (37 C.F.R. § 41.37(c)(1)(iii))

Claims 1-15 were finally rejected in the Final Office Action mailed May 28, 2008. Claims 16-22 were canceled without prejudice on February 21, 2008. Pending claims 1-15 are the subject of this appeal.

The present application includes claims 1-15, which are pending in the present application. Claims 1-4, 7-9, 11-12 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al.

("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), and further in view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"). *See* Final Office Action at pages 5-9 and 11-13.

Claims 5-6 and 13-14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al. ("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), in further view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"), and further in view of U.S. Patent Publication No. 2005/0114909, by Mercier et al. ("Mercier"). *See* Final Office Action at pages 9-10 and 13-14.

Claim 10 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al. ("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), in further view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"), and further in view of Chen, et al., "A Single-Chip MPEG-2 MP@ML Audio/Video Encoder/Decoder with a Programmable Video Interface Unit," IEEE, pp. 941-944, 2001 ("Chen"). *See* Final Office Action at pages 10-11.

The Applicant identifies claims 1-15 as the claims that are being appealed. The text of the pending claims is provided in the Claims Appendix.

## STATUS OF AMENDMENTS
## (37 C.F.R. § 41.37(c)(1)(iv))

The Applicant has not amended any claims subsequent to the final rejection of claims 1-15 mailed on May 28, 2008.

## SUMMARY OF CLAIMED SUBJECT MATTER
## (37 C.F.R. § 41.37(c)(1)(v))

Independent claim 1 recites the following:

A method for producing a high definition video signal[1] comprising:

demuxing a high definition program stream into at least one high definition video data stream component and a plurality of companion component data streams; [2]

muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream; [3]

demuxing the standard definition program stream into a standard definition video data stream, and a subpicture data stream; [4]

scaling the standard definition video stream to a resolution consistent with the high definition video data stream; [5]

overlaying the scaled standard definition video stream with the demuxed subpicture data stream; [6]

---

[1] *See* present application, *e.g.*, at page 2, lines 18-19; Figure 5.
[2] *See id.*, *e.g.*, at page 2, lines 19-21; page 6, line 26 to page 7, line 4; Figure 5 (5080).
[3] *See id.*, *e.g.*, at page 2, lines 22-23; page 7, lines 9-12; Figure 5 (5090).
[4] *See id.*, *e.g.*, at page 2, lines 25-26; page 7, lines 15-17; Figure 5 (5060).
[5] *See id.*, *e.g.*, at page 2, line 27 to page 3, line 2; page 7, lines 19-20; Figure 5 (5120).
[6] *See id.*, *e.g.*, at page 3, lines 2-3; page 7, lines 17-18; Figure 5 (5120).

and replacing the standard definition video stream with the at least one high definition video data stream to produce a high definition video data signal.[7]

Claims 2-10 are dependent upon claim 1.

Independent claim 11 recites the following:

An apparatus for use in producing high a definition video data signal, [8] comprising:

a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one high definition video data stream and a set of other component data streams; [9]

a generator for generating a standard definition video stream; [10]

a muxer for combining the generated standard definition video stream with the set of other component data streams into a standard definition program stream; [11]

a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream; [12]

---

[7] *See* present application, *e.g.*, at page 3, lines 3-5; page 7, line 26 to page 8, line 2; Figure 5 (5130).
[8] *See* present application, *e.g.*, at page 3, lines 11-12; Figure 3 (304); Figure 4 (304).
[9] *See id.*, *e.g.*, at page 3, lines 12-15; page 6, line 26 to page 7, line 4; Figure 4 (412, 414, 416-418).
[10] *See id.*, *e.g.*, at page 3, line 16; page 7, 11-13; Figure 4 (420).
[11] *See id.*, *e.g.*, at page 3, lines 16-18; page 7, lines 9-13; Figure 4 (422).

a video mixer for replacing the scaled up standard definition video stream with the high definition video data stream; [13]

and an encrypter for creating a high definition video data signal from the high definition video data stream and the set of other component data streams. [14]

Claims 12-15 are dependent upon claim 11.

## GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL (37 C.F.R. § 41.37(c)(1)(vi))

Claims 1-4, 7-9, 11-12 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al. ("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), and further in view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"). *See* Final Office Action at pages 5-9 and 11-13.

Claims 5-6 and 13-14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al. ("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), in further view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"), and further in view

---

[12] *See* present application, *e.g.*, at page 3, lines 18-20; page 7, lines 19-20; Figure 4 (430).
[13] *See id.*, *e.g.*, at page 3, lines 20-21; page 7, line 23 to page 8, line 2; Figure 4 (434).
[14] *See id.*, *e.g.*, at page 3, lines 21-23; Figure 4 (436).

of U.S. Patent Publication No. 2005/0114909, by Mercier et al. ("Mercier"). *See* Final Office Action at pages 9-10 and 13-14.

Claim 10 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Publication No. 2001/20038746, by Hughes et al. ("Hughes"), in view of Applicants Admitted Prior Art ("AAPA"), in further view of U.S. Patent Publication No. 2004/0022318, by Garrido et al. ("Garrido"), and further in view of Chen, et al., "A Single-Chip MPEG-2 MP@ML Audio/Video Encoder/Decoder with a Programmable Video Interface Unit," IEEE, pp. 941-944, 2001 ("Chen"). *See* Final Office Action at pages 10-11.

## ARGUMENT
## (37 C.F.R. § 41.37(c)(1)(vii))

In the Final Office Action, claims 1-15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over various combinations of Hughes, Applicants Admitted Prior Art (AAPA), Garrido, Mercier and Chen.

### I. The Proposed Combination of Hughes, Applicants Admitted Prior Art and Garrido Does Not Render Claims 1-4, 7-9, 11-12 and 15 Unpatentable

The Applicant turns to the rejection of claims 1-4, 7-9, 11-12 and 15 as being unpatentable over Hughes in view of Applicants Admitted Prior Art (AAPA) and further in view of Garrido.

#### A. Rejection of Independent Claim 1

With regard to the rejection of independent claim 1 under 103(a), the Applicant submits that the combination of references cited in the Final Office Action fails to disclose, for example, at least the limitations of "[a] method for producing a high definition video signal comprising: demuxing a high definition program stream into at least one high definition video data stream component and a plurality of companion component data streams; muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream; demuxing the standard definition video stream to a resolution consistent with the high definition video data stream; scaling the standard definition video stream to a resolution

consistent with the high definition video data stream; overlaying the scaled standard definition video stream with the demuxed subpicture data stream; and replacing the standard definition video stream with at least one high definition video data stream to produce a high definition video data signal."

With regard to "[a] method for producing a high definition video signal comprising: demuxing a high definition program stream into at least one high definition video data stream component and a plurality of companion component data streams," the Final Office Action alleges that the above claim element is disclosed in Hughes' Fig. 1. (Final Office Action, Page 5). As stated in Hughes, "FIG. 1 illustrates a system that separates a high-resolution source image into a base layer and an enhancement layer, and stores the base layer and the enhancement layer in separate tracks on a storage medium." (Hughes, Paragraph [0015]). Nowhere in Hughes is there any mention of "demuxing a high definition program stream into at least one **high definition video data stream** component and a **plurality of companion component data streams**." Rather, Hughes discloses separating a high-resolution source image into **a** **base layer** and **an** **enhancement layer**. (Hughes, Paragraphs [0027]-[0034]). Further, Hughes discloses that "a standard definition image is generated by decoding the base layer data. A high-resolution image is generated by decoding and combining **both the base layer data and the enhancement layer data**." (Hughes, Paragraph [0008]).

It is unclear what exactly the Examiner is interpreting the "at least one high definition video data stream component and a plurality of companion component data

streams" to be in Hughes. As shown above, the decoding and combination of both the base layer data and the enhancement layer data make up a high-resolution image in Hughes. (Hughes, Paragraph [0008]). Thus, if the Examiner is interpreting the decoding and combination of both the base layer and the enhancement layer to be the "at least one high definition video data stream component," then Hughes fails to disclose "a plurality of companion component data streams." Alternatively, if the Examiner is interpreting the enhancement layer data to be "at least one high definition video data stream component," the Final Office Action: (1) fails to show "a plurality of companion component data streams" because Hughes' base layer is not "a plurality of companion component data streams," and (2) mischaracterizes the Applicant's definition of "component data streams" as set forth in the Applicant's specification (*See e.g.*, Applicant's Specification, Page 5, Lines 17-22 and Page 7, Lines 2-5).

Also, the Applicant notes that storing a base layer and enhancement layer in separate tracks on a storage medium does not "produce a high definition video signal." The Applicant notes that in Hughes, a high-resolution image/stream is not generated until the both the base layer data and the enhancement layer data are decoded and combined. (Hughes, Paragraph [0008]). Thus, Hughes' disclosed method of storing a base layer and enhancement layer in separate tracks on a storage medium does not teach "[a] method for producing a high definition video signal," as recited in Applicant's independent claim 1.

With regard to "muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream" **and then** "demuxing the standard definition video stream to a resolution consistent with the high definition video data stream," the Final Office Action alleges that the above claim elements are disclosed in AAPA Fig. 2. (Final Office Action, Page 6). However, AAPA Fig. 2 discloses (Step 1) decrypting the program stream; (Step 2) separating the program stream into a standard definition video stream component, a compressed audio stream component, a compressed subpicture stream component and a navigational stream component; (Step 3) sending the compressed video stream component to a video decompression device, the compressed subpicture stream component to a subpicture decode device, the compressed audio stream component to an audio decompression device and the navigational stream component to a system control processor; and (Step 4) mixing the decompressed video and decoded subpicture streams at a video mixer and sent to a standard definition television for viewing while the decompressed audio stream is sent to an audio receiver for playback. (See Applicant's Specification, Page 5, Line 17 to Page 6, Line 11).

It is unclear what exactly the Examiner is interpreting the "muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream" **and then** "demuxing the standard definition video stream to a resolution consistent with the high definition video data stream," to be in AAPA Fig. 2. First, nowhere in AAPA Fig. 2 is there any disclosure regarding muxing

and then demuxing. Second, if the Examiner is interpreting AAPA's disclosure of mixing the decompressed video stream and decoded subpicture stream at a video mixer to be "muxing **the plurality of companion component data streams with a standard resolution video stream** into a standard definition video program stream," the Applicant notes that the decoded subpicture stream is not **a plurality of companion component data streams**.

Further, the Final Office Action fails to show a motivation to combine Hughes' Fig. 1 with AAPA Fig. 2. As discussed above, Hughes' Fig. 1 discloses "a system that separates **a high-resolution source image** into a base layer and an enhancement layer, and **stores the base layer and the enhancement layer in separate tracks on a storage medium**." (Hughes, Paragraph [0015]). AAPA is unrelated to storage of a base layer and enhancement layer on a storage medium and does not receive or deal with high-resolution source images. Rather, AAPA Fig. 2 discloses decrypting, demuxing, decompressing, decoding, mixing and displaying **a standard definition program stream**. With regard to the separating the program stream into four components as illustrated in AAPA Fig. 2, Hughes fails to discuss a subpicture stream component, an audio stream component and a navigational stream component. Rather, Hughes merely discusses video information (e.g., base layer data is decoded to generate a standard definition image; and the base layer data and the enhancement layer data is decoded and combined to generate a high-resolution image). AAPA Fig. 2 does not teach separating the standard definition compressed video stream into a base

layer and enhancement layer. In fact, such separation would not be possible in AAPA Fig. 2 because in AAPA Fig. 2, a standard definition program stream is received instead of the high-resolution video stream received in Hughes. Thus, the Examiner has failed to make a *prima facie* case of obviousness because the Examiner has not made a clear articulation of the reason(s) why the claimed invention would have been obvious. Instead, the Examiner bases her rejection on mere conclusory statements instead of some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. (*See* the MPEP at § 2142).

With regard to "scaling the standard definition video stream to a resolution consistent with the high definition video data stream" and "overlaying the scaled standard definition video stream with the demuxed subpicture data stream," the Final Office Action alleges that the above claim elements are disclosed in Garrido's Paragraph [0037]. (Final Office Action, Page 7). However, the Applicant initially notes that nowhere in Garrido's Paragraph [0037] is there any mention of "overlaying the scaled standard definition video stream with the demuxed subpicture data stream."

Further, the Applicant maintains that (1) Hughes teaches away from the combination with Garrido, and (2) modifying Hughes with Garrido, as proposed by the Final Office Action, would render Hughes inoperable for its intended purpose. The Response to Arguments section states that "Hughes discloses the base layer and the enhancement layer are decoded simultaneously [0013]. Since Hughes discloses to

generate a high definition signal by combining both the base and enhancement layer data, it is clear to the examiner that Hughes would obviously include scaling the standard definition signal." (Final Office Action, Page 3, Lines 6-10). However, Hughes discloses for a standard definition display, "[a] DVD reader reads the base layer data from the default camera angle track of the DVD (step 222). The base layer data is then decoded (step 224). The decoded base layer data is displayed on a standard definition display (step 226), thereby recreating the original sequence of images." (Hughes, Paragraph [0040], Lines 2-7). **The Applicant notes that for standard definition display, there is no need to "scal[e] the standard definition video stream to a resolution consistent with the high definition video data stream" because the stream is being displayed on a standard definition display**.

Alternatively, Hughes discloses for a high-resolution display, decoding and combining both the base layer and the enhancement layer. (Hughes, Paragraph [0008], Lines 8-10 and Paragraphs [0042]-[0045]). **The Applicant notes that because the decoding and combination of the base layer and the enhancement layer generates a high resolution stream, there is no standard definition video stream to scale**. Combining the enhancement layer and base layer is different than scaling a standard definition video stream. If the Final Office Action is interpreting "base layer data" to be "a standard definition video stream," the Applicant notes that Hughes' discloses that "a standard definition image is generated **by decoding** the base layer

data." (Hughes, Paragraph [0008]). In other words, the base layer data itself is not a standard definition video stream.

The Response to Arguments section further states that "[i]t is clear to the examiner that it would be obvious to scales the base layer in Hughes to generate the high definition signal." However, as discussed above, the base layer itself is not a standard definition video stream so it makes no sense to scale the base layer in Hughes. Further, Hughes discloses "a system...that allows both a standard definition version of a video program and a high-resolution version of the same program to be efficiently stored on a single DVD...." (Hughes, Paragraph [0007]). Because the DVD in Hughes stores both a standard definition version and a high-resolution version, it does not make sense to scale the standard definition version when a high-resolution version is already stored and available on the same DVD. Thus, the disclosure of Hughes teaches away from "scaling the standard definition video stream to a resolution consistent with the high definition video data stream," and adding scaling as taught in Garrido would render Hughes inoperable for its intended purpose. (*See also*, Non-Final Office Action Response filed February 21, 2008, Arguments on Pages 5-7).

With regard to "replacing the standard definition video stream with at least one high definition video data stream to produce a high definition video data signal," the Final Office Action alleges that the above claim element is disclosed in Hughes' Paragraph [0044]. (Final Office Action, Pages 5-6). However, Hughes' Paragraph

[0044] states that "[t]he outputs of decompressor 304 and decompressor 306 are coupled to a decoding and combining module 308, which decodes and combines the base layer data with the enhancement layer data to generate a high-resolution display 310." Hughes' teaching of combining base layer data with enhancement layer data is not the same as "replacing the standard definition video stream with the at least one high definition video data stream to produce a high definition video data signal," as set forth in Applicant's independent claim 1.

Therefore, the Applicant maintains that at least the limitations "[a] method for producing a high definition video signal comprising: demuxing a high definition program stream into at least one high definition video data stream component and a plurality of companion component data streams; muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream; demuxing the standard definition video stream to a resolution consistent with the high definition video data stream; scaling the standard definition video stream to a resolution consistent with the high definition video data stream; overlaying the scaled standard definition video stream with the demuxed subpicture data stream; and replacing the standard definition video stream with at least one high definition video data stream to produce a high definition video data signal," as recited by the Applicant in independent claim 1, are not obvious over Hughes in view of AAPA and

further in view of Garrido. Accordingly, independent claim 1 is not unpatentable over Hughes in view of AAPA and further in view of Garrido and is allowable.

**B. Rejection of Dependent Claims 2-4 and 7-9**

Claims 2-4 and 7-9 depend directly or indirectly on independent claim 1. Therefore, the Applicant submits that claims 2-4 and 7-9 are allowable over the combination of references cited in the Final Office Action at least for the reasons stated above with regard to claim 1.

The Applicant also submits that at least the limitation of "determining if the received program data stream is a high definition program data stream," as recited by the Applicant in claim 3; and "generating the standard resolution video stream," as recited by the Applicant in claim 9, are not obvious over Hughes in view of AAPA and further in view of Garrido.

With regard to claim 3, the Final Office Action states the following at page 7:

> Regarding claim 3, the combination of Hughes, AAPA, and Garrido as a whole further teaches everything as claimed above, see claim 1. In addition, Hughes teaches the method of claim 2 further including determining if the received program data stream is a high definition program data stream (Hughes teaches he decoding and combining module may generate an encoded high definition MPEG-2 stream. Further, Hughes discloses the output of base layer may be coupled to a standard definition display device for displaying the video content at a standard resolution ([0045]), which reads on the claimed limitation. Further, it is clear to the examiner that the method as disclosed by Hughes would necessitate determining the type of program stream received in-order to properly display the content.

(Final Office Action at page 7). However, the Applicant notes that Hughes only teaches receiving high resolution source images. (Hughes, Figure 1 (100), Paragraph [0027]). Nothing in Hughes indicates that Hughes' encoding system receives anything other than high resolution source images. Further, if the Examiner is alleging that Hughes teaches determining whether to generate and display in high resolution or standard resolution, the Applicant notes that such disclosure is different than Applicant's dependent claim 3 because determining whether to display in high resolution or standard resolution is different than "determining if **the received program data stream** is a high definition program data stream" as recited in Applicant's dependent claim 3. Accordingly, the Applicant submits that claim 3 is allowable over the combination of references cited in the Final Office Action at least for the above reasons

With regard to claim 9, the Final Office Action states the following at page 7:

> Regarding claim 9, the combination of Hughes and AAPA teaches everything as claimed above, see claim 1. In addition, Hughes teaches the method of claim 1 further comprising generating the standard resolution data stream ([0045]).

(Final Office Action at page 9). The Applicant notes that the cited section of Hughes merely discloses the decoding system's 300 base layer decompressor 304 being coupled to a standard definition display device for displaying the video content at a standard resolution. (Hughes, Paragraph [0045]). However, Applicant's dependent claim 9 depends from Applicant's independent claim 1, which with regard to the "standard resolution video stream" recites "muxing the plurality of companion

component data streams with a standard resolution video stream into a standard definition video program stream." Clearly, the "standard resolution video stream" set forth in Applicant's independent claim 1 and dependent claim 9 is different than Hughes' base layer decompressor 304 output. Accordingly, the Applicant submits that claim 9 is allowable over the combination of references cited in the Final Office Action at least for the above reasons.

The Applicant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 2-4 and 7-9.

### C. Rejection of Independent Claim 11

With regard to the rejection of independent claim 1 under 103(a), the Applicant submits that the combination of references cited in the Final Office Action fails to disclose, for example, at least the limitations of "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one high definition video data stream and a set of other component data streams; a generator for generating a standard definition video stream; a muxer for combining the generated standard definition video stream with the set of other component data streams into a standard definition program stream; a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream; a video mixer for replacing the scaled up standard definition video stream with

the high definition video data stream; and an encrypter for creating a high definition video data signal from the high definition video data stream and the set of other component data streams."

With regard to "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one high definition video data stream and a set of other component data streams," the Final Office Action alleges that the above claim element is disclosed in Hughes' Fig. 1. (Final Office Action, Page 11). The Applicant notes that in Hughes Fig. 1, there is no disclosure regarding "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream...." Rather, Hughes' Fig. 1 separates a high-resolution source image into a base layer and an enhancement layer, and stores the base layer and the enhancement layer in separate tracks on a storage medium. (Hughes, Paragraph [0015]). Neither the base layer nor the enhancement layer is a component data stream. Rather, the base layer and enhancement layer is data that once decoded and combined, generate a high resolution image/stream. (Hughes, Paragraph [0008]). The Applicant notes, however, that such decoding and combination does not occur in Fig. 1. Rather, Fig. 1 is related to storing the base layer and enhancement layer in separate tracks on a storage medium. Thus, Hughes' Fig. 1 cannot disclose "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams

comprising at least one high definition video data stream and a set of other component data streams," as set forth in Applicant's independent claim 11.

Additionally, as stated in Hughes, "FIG. 1 illustrates a system that separates a high-resolution source image into a base layer and an enhancement layer, and stores the base layer and the enhancement layer in separate tracks on a storage medium." (Hughes, Paragraph [0015]). Nowhere in Hughes is there any mention of "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one **high definition video data stream** and **a set of other component data streams**." Rather, Hughes discloses separating a high-resolution source image into **a base layer** and **an enhancement layer**. (Hughes, Paragraphs [0027]-[0034]). Further, Hughes discloses that "a standard definition image is generated by decoding the base layer data. A high-resolution image is generated by decoding and combining **both the base layer data and the enhancement layer data**." (Hughes, Paragraph [0008]).

It is unclear what exactly the Examiner is interpreting the "at least one high definition video data stream and a set of other component data streams" to be in Hughes. As shown above, the decoding and combination of both the base layer data and the enhancement layer data make up a high-resolution image in Hughes. Thus, if the Examiner is interpreting the decoding and combination of both the base layer and the enhancement layer to be the "at least one high definition video data stream," then

Hughes fails to disclose "a set of other component data streams." Alternatively, if the Examiner is interpreting the enhancement layer data to be "at least one high definition video data stream," the Final Office Action: (1) fails to show "a set of other component data streams" because Hughes' base layer is not "a set of other component data streams," and (2) mischaracterizes the Applicant's definition of "component data streams" as set forth in the Applicant's specification (*See e.g.*, Applicant's Specification, Page 5, Lines 17-23 and Page 7, Lines 2-5).

Also, the Final Office Action further states that "it is clear to the examiner, that the high definition stream would necessitate the component data streams, as the stream is recorded on a DVD as disclosed by Hughes." (Final Office Action, Page 11, Lines 11-13). Based on the comments in the Final Office Action, it appears as though the Examiner acknowledges that Hughes fails to explicitly teach the claimed element and instead alleges that, with regard to the "a set of other component data streams," the high definition program stream demuxer for extracting a set of other component data streams from a high definition stream is an inherent feature of Hughes.

The Applicants submit that a rejection based on inherency must include a statement of the rationale or evidence tending to show inherency. *See* Manual of Patent Examining Procedure at § 2112. "The fact that a certain result or characteristic <u>may</u> occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic." *See id. citing In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993).

> To establish inherency, the extrinsic evidence "must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.

*In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999). The Applicants respectfully submit that neither Hughes itself nor the Office Action "make[s] clear that the missing descriptive matter," said to be inherent "is necessarily present in" Hughes.

A rejection based on inherency must be based on factual or technical reasoning:

> In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teaching of the applied prior art.

*Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990).

The Applicants respectfully submit that the Final Office Action does not contain a basis in fact and/or technical reasoning to support the rejection based on inherency. Instead, as recited above, at least claim 11 of the present application stands rejected based on a conclusory statement of inherency, rather than upon a "basis in fact and/or technical reasoning." Accordingly, the Applicants respectfully submit that, absent a "basis in fact and/or technical reasoning" for the rejection of record, that rejection should be reversed.

With regard to "a generator for generating a standard definition video stream," the Final Office Action alleges that the above claim element is disclosed in Hughes' Fig. 1, 104, base layer generator. The Applicant notes that "[t]he base layer generator 104 generates a base layer portion of the source image 100 and communicates the base layer to a compressor 108." (Hughes, Paragraph [0029]). As discussed above, a base layer is different than a standard definition video stream in that the base layer data needs to be decoded to generate a standard definition image/stream. (Hughes, Paragraph [0008]). Thus, Hughes' base layer generator 104 is different than "a generator for generating a standard definition video stream," as recited in Applicant's independent claim 11.

With regard to "a muxer for combining the generated standard definition video stream with the set of other component data streams into a standard definition program stream," the Final Office Action alleges that the above claim element is disclosed in AAPA Fig. 2. The Applicant notes that "[t]he base layer generator 104 generates a base layer portion of the source image 100 and communicates the base layer to a compressor 108." (Hughes, Paragraph [0029]). As discussed above, a base layer is different than a standard definition video stream in that the base layer data needs to be decoded to generate a standard definition image/stream. (Hughes, Paragraph [0008]). Thus, Hughes' base layer generator 104 is different than "a generator for generating a standard definition video stream," as recited in Applicant's independent claim 11. As

discussed above with regard to independent claim 1, if the Final Office Action is interpreting AAPA's disclosure of mixing the decompressed video stream and decoded subpicture stream at a video mixer to be "a muxer for combining the generated standard definition video stream with the **set of other component data streams** into a standard definition program stream," the Applicant notes that the decoded subpicture stream is not **a set of other component data streams**.

Further, as mentioned above, the Final Office Action fails to show a motivation to combine Hughes' Fig. 1 with AAPA Fig. 2. Hughes' Fig. 1 discloses "a system that separates **a high-resolution source image** into a base layer and an enhancement layer, and **stores the base layer and the enhancement layer in separate tracks on a storage medium**." (Hughes, Paragraph [0015]). AAPA is unrelated to storage of a base layer and enhancement layer on a storage medium and does not receive or deal with high-resolution source images. Rather, AAPA Fig. 2 discloses decrypting, demuxing, decompressing, decoding, mixing and displaying a standard definition program stream. With regard to the separating the program stream into four components as illustrated in AAPA Fig. 2, Hughes fails to discuss a subpicture stream component, an audio stream component and a navigational stream component. Rather, Hughes merely discusses video information (e.g., base layer data is decoded to generate a standard definition image; and the base layer data and the enhancement layer data is decoded and combined to generate a high-resolution image). AAPA Fig. 2 does not teach separating the standard definition compressed video stream into a base

layer and enhancement layer. In fact, such separation would not be possible in AAPA Fig. 2 because in AAPA Fig. 2, a standard definition program stream is received instead of the high-resolution video stream received in Hughes. Thus, the Final Office Action has failed to make a *prima facie* case of obviousness because the Examiner has not made a clear articulation of the reason(s) why the claimed invention would have been obvious. Instead, the Examiner bases her rejection on mere conclusory statements instead of some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. (*See* the MPEP at § 2142).

With regard to "a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream," the Final Office Action alleges that the above claim element is disclosed in Garrido's Paragraph [0037]. However, the Applicant maintains that (1) Hughes teaches away from the combination with Garrido, and (2) modifying Hughes with Garrido, as proposed by the Final Office Action, would render Hughes inoperable for its intended purpose. The Response to Arguments section states that "Hughes discloses the base layer and the enhancement layer are decoded simultaneously [0013]. Since Hughes discloses to generate a high definition signal by combining both the base and enhancement layer data, it is clear to the examiner that Hughes would obviously include scaling the standard definition signal." (Final Office Action, Page 3, Lines 6-10). However, Hughes discloses for a standard definition display, "[a] DVD reader reads the base layer data

from the default camera angle track of the DVD (step 222). The base layer data is then decoded (step 224). The decoded base layer data is displayed on a standard definition display (step 226), thereby recreating the original sequence of images." (Hughes, Paragraph [0040], Lines 2-7). **The Applicant notes that for standard definition display, there is no need for a "video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream" because the stream in Hughes is being displayed on a standard definition display**.

Alternatively, Hughes discloses for a high-resolution display, decoding and combining both the base layer and the enhancement layer. (Hughes, Paragraph [0008], Lines 8-10 and Paragraphs [0042]-[0045]). **The Applicant notes that because the decoding and combination of the base layer and the enhancement layer generates a high resolution stream, there is no standard definition video stream to scale**. Combining the enhancement layer and base layer is different than scaling a standard definition video stream. If the Examiner is interpreting "base layer data" to be "a standard definition video stream," the Applicant notes that Hughes' discloses that "a standard definition image is generated **by decoding** the base layer data." (Hughes, Paragraph [0008]). In other words, the base layer data itself is not a standard definition video stream.

The Response to Arguments section of the Final Office Action further states that "[i]t is clear to the examiner that it would be obvious to scales the base layer in Hughes

to generate the high definition signal." (Final Office Action, Page 4). However, as discussed above, the base layer itself is not a standard definition video stream so it makes no sense to scale the base layer in Hughes. Further, Hughes discloses "a system...that allows both a standard definition version of a video program and a high-resolution version of the same program to be efficiently stored on a single DVD...." (Hughes, Paragraph [0007]). Because the DVD in Hughes stores both a standard definition version and a high-resolution version, it does not make sense to scale the standard definition version when a high-resolution version is already stored and available on the same DVD. Thus, the disclosure of Hughes teaches away from "a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream," and adding scaling as taught in Garrido would render Hughes inoperable for its intended purpose. (*See also*, Non-Final Office Action Response filed February 21, 2008, Arguments on Pages 5-7).

With regard to "a video mixer for replacing the scaled up standard definition video stream with the high definition video data stream," the Final Office Action alleges that the above claim element is disclosed in Hughes' Paragraph [0044]. (Final Office Action, Page 11). However, Hughes' Paragraph [0044] states that "[t]he outputs of decompressor 304 and decompressor 306 are coupled to a decoding and combining module 308, which decodes and combines the base layer data with the enhancement layer data to generate a high-resolution display 310." Hughes' teaching of combining

base layer data with enhancement layer data using a decoding and combining module is not the same as "a video mixer for **<u>replacing</u> the scaled up standard definition video stream with the high definition video data stream**." As mentioned above, Hughes teaches either (1) generating a standard definition image by decoding the base layer data if for a standard definition display, **<u>or</u>** (2) generating a high-resolution image by decoding and combining both the base layer data and the enhancement layer data if for a high-resolution display. Nowhere in Hughes is there any disclosure of **<u>replacing</u> the scaled up standard definition video stream with the high definition video data stream**," as set forth in Applicant's independent claim 11.

With regard to "an encrypter for creating a high definition video data signal from the high definition video data stream and the set of other component data streams," the Final Office Action alleges that the above claim element is disclosed in Garrido's Paragraph [0063] and further states that "it is clear to the examiner since Garrido discloses encrypting the video, it would be necessitate the use of an encrypter." (Final Office Action, Page 13, Lines 1-2). However, Garrido's Paragraph [0063], in its entirety, states that "[c]lassification also forces unimportant codevectors that do not strongly fall into any class to merge with like codevectors." (Garrido, Paragraph [0063]). Even if Garrido's Paragraph [0063] necessitated the use of an encrypter as alleged by the Final Office Action (which it does not), the combination of references still fails to disclose "an encrypter **<u>for creating a high definition video data signal from the high definition</u>**

**video data stream and the set of other component data streams**." As mentioned above, the Final Office Action has failed to make a *prima facie* case of obviousness because the Examiner has not made a clear articulation of the reason(s) why the claimed invention would have been obvious. Instead, the Examiner bases her rejection on mere conclusory statements instead of some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness. (*See* the MPEP at § 2142).

Therefore, the Applicant maintains that at least the limitations "a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one high definition video data stream and a set of other component data streams; a generator for generating a standard definition video stream; a muxer for combining the generated standard definition video stream with the set of other component data streams into a standard definition program stream; a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream; a video mixer for replacing the scaled up standard definition video stream with the high definition video data stream; and an encrypter for creating a high definition video data signal from the high definition video data stream and the set of other component data streams," as recited by the Applicant in independent claim 11, are not obvious over Hughes in view of AAPA and further in view of Garrido.

Accordingly, independent claim 11 is not unpatentable over Hughes in view of AAPA and further in view of Garrido and is allowable.

### D. Rejection of Dependent Claims 12 and 15

Claims 12 and 15 depend on independent claim 11. Therefore, the Applicant submits that claims 12 and 15 are allowable over the reference cited in the Final Office Action at least for the reasons stated above with regard to claim 11.

The Applicant also submits that at least the limitation of "a receiver for receiving a program data stream," as recited by the Applicant in claim 12; and "a router for determining if the received program data stream is a high definition program stream," as recited by the Applicant in claim 15, is not obvious over Hughes in view of AAPA and further in view of Garrido.

With regard to claim 12, the Final Office Action states the following at page 5:

> Regarding claim 12, the combination of Hughes, AAPA and Garrido teaches everything as claimed above, see claim 11. In addition, Hughes teaches the apparatus of claim 11 further including a receiver for receiving a program data stream (storage medium, DVD, fig. 1).

(Final Office Action at page 13). The Applicant notes that a receiver is different than a storage medium and one skilled in the art would not confuse the two. Accordingly, the Applicant submits that claim 12 is allowable over the combination of references cited in the Final Office Action at least for the above reasons.

With regard to claim 15, the Final Office Action states the following at page 13:

> Regarding claim 15, the combination of Hughes, AAPA and Garrido teaches everything as claimed above, see claim 11. In addition, Hughes further teaches the apparatus of claim 12 further including a router for determining if the received program data stream is a high definition program stream ([0045]).

(Final Office Action at page 13). The Applicant notes that Hughes' Paragraph [0045] states the following:

> Alternatively, the decoding and combining module 308 may generate an encoded high-definition MPEG-2 stream (or transcode to another encoded format), or could provide the decoded video to a distribution device (not shown) for transmission to remote devices. Although not shown in FIG. 5, the output of base layer decompressor 304 may also be coupled to a standard definition display device for displaying the video content at a standard resolution.

(Hughes, Paragraph [0045]). Clearly, nowhere in the cited paragraph of Hughes is there any mention of a router, let alone "a router for determining if the received program data stream is a high definition program stream," as recited in Applicant's dependent claim 15. Accordingly, the Applicant submits that claim 15 is allowable over the combination of references cited in the Final Office Action at least for the above reasons.

The Applicant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 12 and 15.

## II. The Proposed Combination of Hughes, Applicants Admitted Prior Art, Garrido and Mercier Does Not Render Claims 5-6 and 13-14 Unpatentable

The Applicant turns to the rejection of claims 5-6 and 13-14 as being unpatentable over Hughes in view of AAPA, in further view of Garrido and further in view of Mercier.

Claims 5-6 and 13-14 depend on independent claims 1 and 11, respectively, and Mercier fails to remedy the previously mentioned deficiencies of Hughes in view of AAPA and further in view of Garrido. Therefore, the Applicant submits that claims 5-6 and 13-14 are allowable over the combination of references cited in the Final Office Action at least for the reasons stated above with regard to claims 1 and 11.

Accordingly, the Applicant submits that claims 5-6 and 13-14 are allowable over the combination of references cited in the Final Office Action at least for the above reasons. The Applicant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claims 5-6 and 13-14.

### III. The Proposed Combination of Hughes, Applicants Admitted Prior Art, Garrido and Chen Does Not Render Claim 10 Unpatentable

The Applicant turns to the rejection of claim 10 as being unpatentable over Hughes in view of AAPA, in further view of Garrido and further in view of Chen.

Claims 10 depends on independent claim 1 and Chen fails to remedy the previously mentioned deficiencies of Hughes in view of AAPA and further in view of Garrido. Therefore, the Applicant submits that claim 10 is allowable over the combination of references cited in the Final Office Action at least for the reasons stated above with regard to claim 1. The Applicant further notes that the Chen reference provided to the Applicant by the Examiner (attached as Evidence Exhibit 4) has a blank Page 943 (i.e., the reference is missing the sections between 3.4 and 4.2). Thus, the

Examiner has failed to provide the Applicant with cited sections of Chen relied on in the Final Office Action (i.e., section 4.1).

Accordingly, the Applicant submits that claim 10 is allowable over the combination of references cited in the Final Office Action at least for the above reasons. The Applicant also reserves the right to argue additional reasons beyond those set forth above to support the allowability of claim 10.

## CONCLUSION

For at least the foregoing reasons, the Applicant submits that claims 1-15 are in condition for allowance. Reversal of the Examiner's rejection and issuance of a patent on the application are therefore requested.

The Commissioner is hereby authorized to charge $540 (to cover the Brief on Appeal Fee) and any additional fees or credit any overpayment to the deposit account of McAndrews, Held & Malloy, Ltd., Account No. 13-0017.

Respectfully submitted,

Date: 24-NOV-2008

By: /Philip Henry Sheridan/
Philip Henry Sheridan
Reg. No. 59,918
Attorney for Applicant

McANDREWS, HELD & MALLOY, LTD.
500 West Madison Street, 34th Floor
Chicago, Illinois 60661
(T) 312 775 8000
(F) 312 775 8100

(PHS)

**CLAIMS APPENDIX**
**(37 C.F.R. § 41.37(c)(1)(viii))**

1. A method for producing a high definition video signal comprising:

demuxing a high definition program stream into at least one high definition video data stream component and a plurality of companion component data streams;

muxing the plurality of companion component data streams with a standard resolution video stream into a standard definition video program stream;

demuxing the standard definition program stream into a standard definition video data stream, and a subpicture data stream;

scaling the standard definition video stream to a resolution consistent with the high definition video data stream;

overlaying the scaled standard definition video stream with the demuxed subpicture data stream;

and replacing the standard definition video stream with the at least one high definition video data stream to produce a high definition video data signal.

2. The method of claim 1 further including, prior demuxing the high definition program stream, receiving a program data stream.

3. The method of claim 2 further including determining if the received program data stream is a high definition program data stream.

4. The method of claim 1 wherein the plurality of companion component data streams comprises one or more of audio data stream, a subpicture data stream, and a navigational data stream.

5. The method of claim 1 wherein the high definition program stream is in encrypted format.

6. The method of claim 5 further comprising, prior to demuxing the high definition program stream, decrypting the encrypted high definition program stream.

7. The method of claim 1 wherein the at least one high definition video data stream component is in compressed format.

8. The method of claim 7 further comprising, prior to the replacing step, decompressing the high definition video data stream.

9. The method of claim 1 further comprising generating the standard resolution video stream.

10. The method of claim 9 wherein the generated standard resolution video stream comprises a blue screen video elementary stream.

11. An apparatus for use in producing high a definition video data signal, comprising:

a high definition program stream demuxer for extracting a plurality of component data streams from a high definition program stream, the plurality of component data streams comprising at least one high definition video data stream and a set of other component data streams;

a generator for generating a standard definition video stream;

a muxer for combining the generated standard definition video stream with the set of other component data streams into a standard definition program stream;

a video scaler for increasing the resolution of the standard definition video stream to a resolution consistent with the high definition video stream;

a video mixer for replacing the scaled up standard definition video stream with the high definition video data stream;

and an encrypter for creating a high definition video data signal from the high definition video data stream and the set of other component data streams.

12. The apparatus of claim 11 further including a receiver for receiving a program data stream.

13. The apparatus of claim 12 wherein the received program data stream is in encrypted format.

14. The apparatus of claim 13 further including a decrypter for decrypting the encrypted program data stream.

15. The apparatus of claim 12 further including a router for determining if the received program data stream is a high definition program stream.

## EVIDENCE APPENDIX

### (37 C.F.R. § 41.37(c)(1)(ix))

(1) United States Patent Publication No. 2001/0038746 ("Hughes"), entered into record by the Examiner in the November 21, 2007 Office Action.

(2) United States Patent Publication No. 2004/0022318 ("Garrido"), entered into record by the Examiner in the November 21, 2007 Office Action.

(3) United States Patent Publication No. 2005/0114909 ("Mercier"), entered into record by the Examiner in the November 21, 2007 Office Action.

(4) Chen, et al., "A Single-Chip MPEG-2 MP@ML Audio/Video Encoder/Decoder with a Programmable Video Interface Unit," IEEE, pp. 941-944, 2001 ("Chen"), entered into record by the Examiner in the November 21, 2007 Office Action.

## RELATED PROCEEDINGS APPENDIX
## (37 C.F.R. § 41.37(c)(1)(x))

The Appellant is unaware of any related appeals or interferences.

US 20010038746A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2001/0038746 A1

(43) Pub. Date: Nov. 8, 2001

Hughes, JR. et al.

(54) LAYERED CODING OF IMAGE DATA USING SEPARATE DATA STORAGE TRACKS ON A STORAGE MEDIUM

(76) Inventors: **Robert K. Hughes JR.**, Shoreline, WA (US); **James H. Taylor**, Mountlake Terrace, WA (US)

Correspondence Address:
**LEE & HAYES PLLC**
**421 W RIVERSIDE AVENUE SUITE 500**
**SPOKANE, WA 99201**

(21) Appl. No.: **09/756,824**

(22) Filed: **Jan. 8, 2001**

**Related U.S. Application Data**

(63) Continuation of application No. 09/565,731, filed on May 5, 2000.

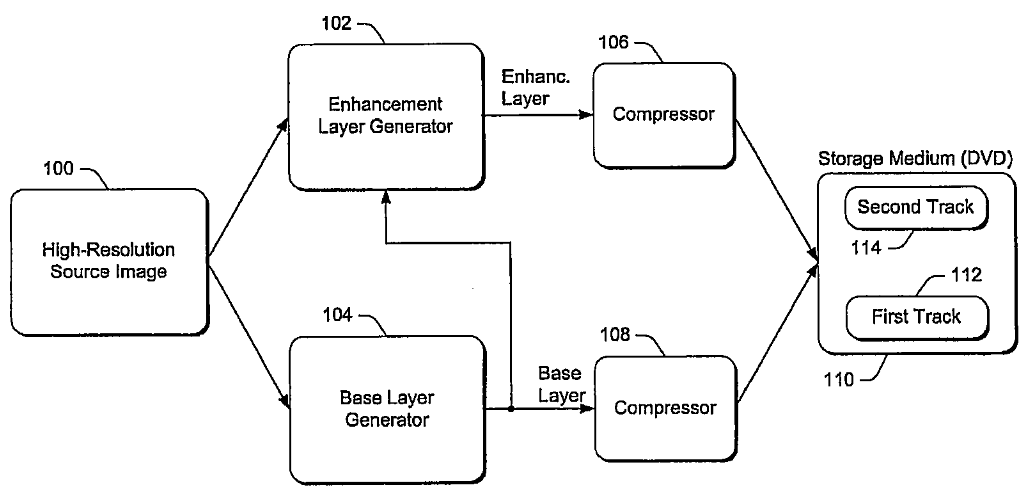
**Publication Classification**

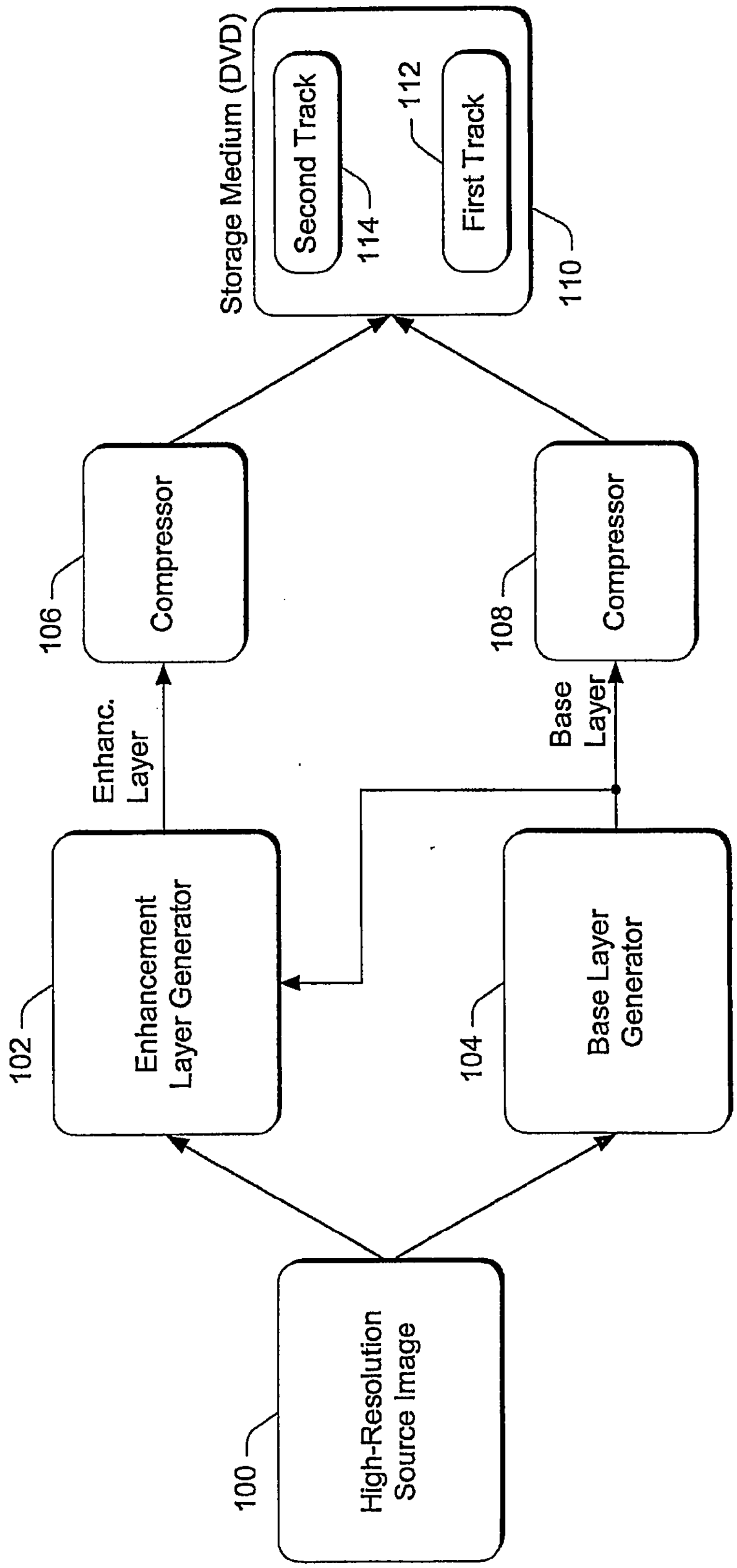
(51) **Int. Cl.**$^7$ .............................. **H04N 5/91**; H04N 5/781

(52) **U.S. Cl.** .............................................. **386/123**; 386/125

(57) **ABSTRACT**

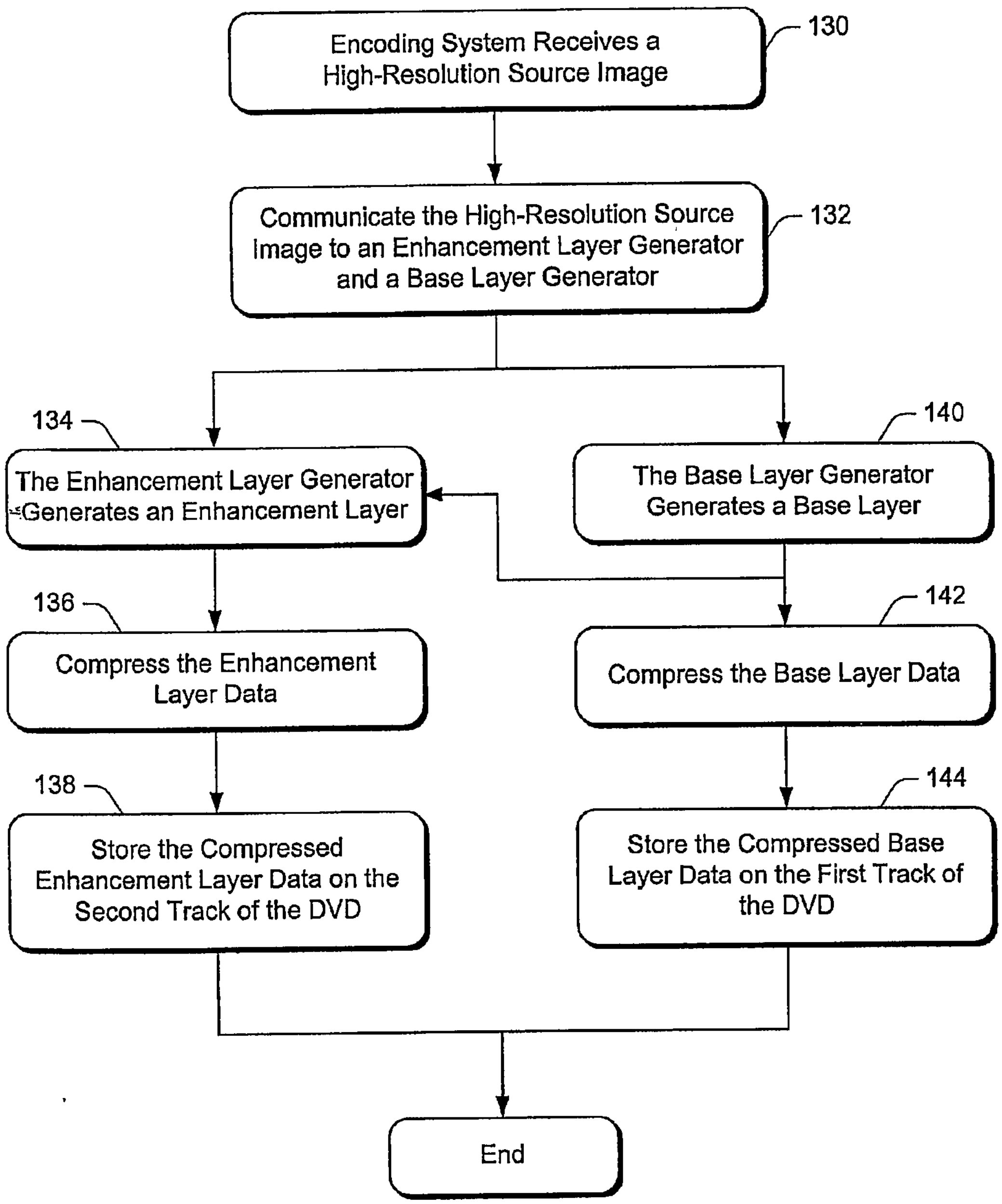
A source image is encoded into a base layer and an enhancement layer. The base layer represents a standard definition portion of the source image and the enhancement layer represents a high-resolution portion of the source image. The base layer is stored on a first data storage track of a storage medium, such as a DVD, and the enhancement layer is stored on a second data storage track of the storage medium. The first data storage track may be a default camera angle track and then second data storage track may be a second camera angle track. The data is formatted such that a standard definition device will not read the enhancement layer data. A high-resolution decoding system decodes the base layer and the enhancement layer simultaneously to generate a high-resolution image.





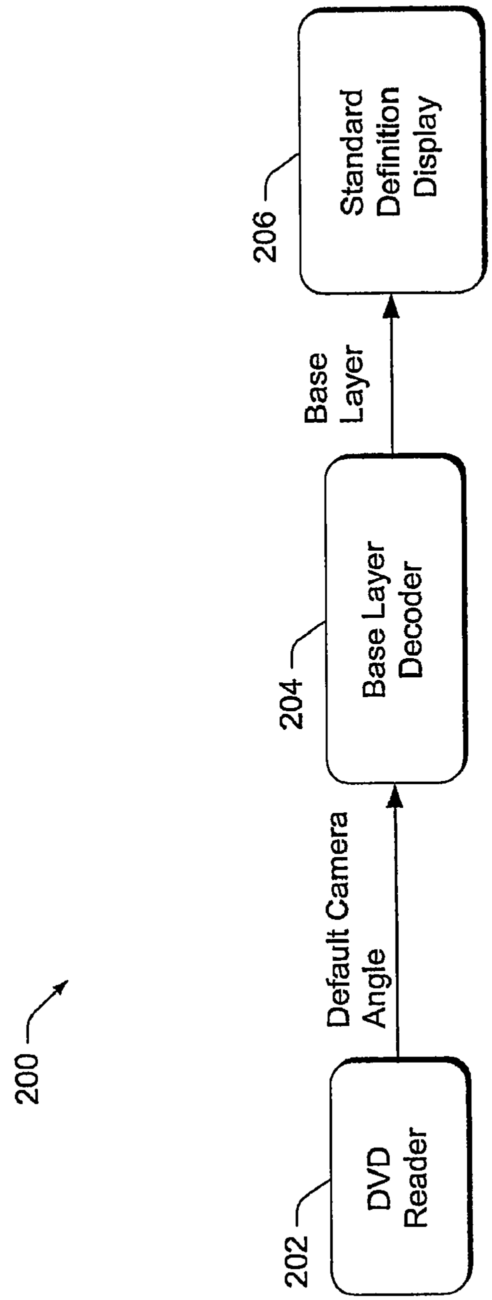
Storage Medium (DVD)
Second Track
114
112
First Track
110
106
Compressor
108
Compressor
Enhanc. Layer
Base Layer
102
Enhancement Layer Generator
104
Base Layer Generator
100
High-Resolution Source Image


Fig. 1



Encoding System Receives a High-Resolution Source Image
130
Communicate the High-Resolution Source Image to an Enhancement Layer Generator and a Base Layer Generator
132
134
The Enhancement Layer Generator Generates an Enhancement Layer
140
The Base Layer Generator Generates a Base Layer
136
Compress the Enhancement Layer Data
142
Compress the Base Layer Data
138
Store the Compressed Enhancement Layer Data on the Second Track of the DVD
144
Store the Compressed Base Layer Data on the First Track of the DVD
End


Fig. 2



200
202
DVD Reader
Default Camera Angle
204
Base Layer Decoder
Base Layer
206
Standard Definition Display


Fig. 3

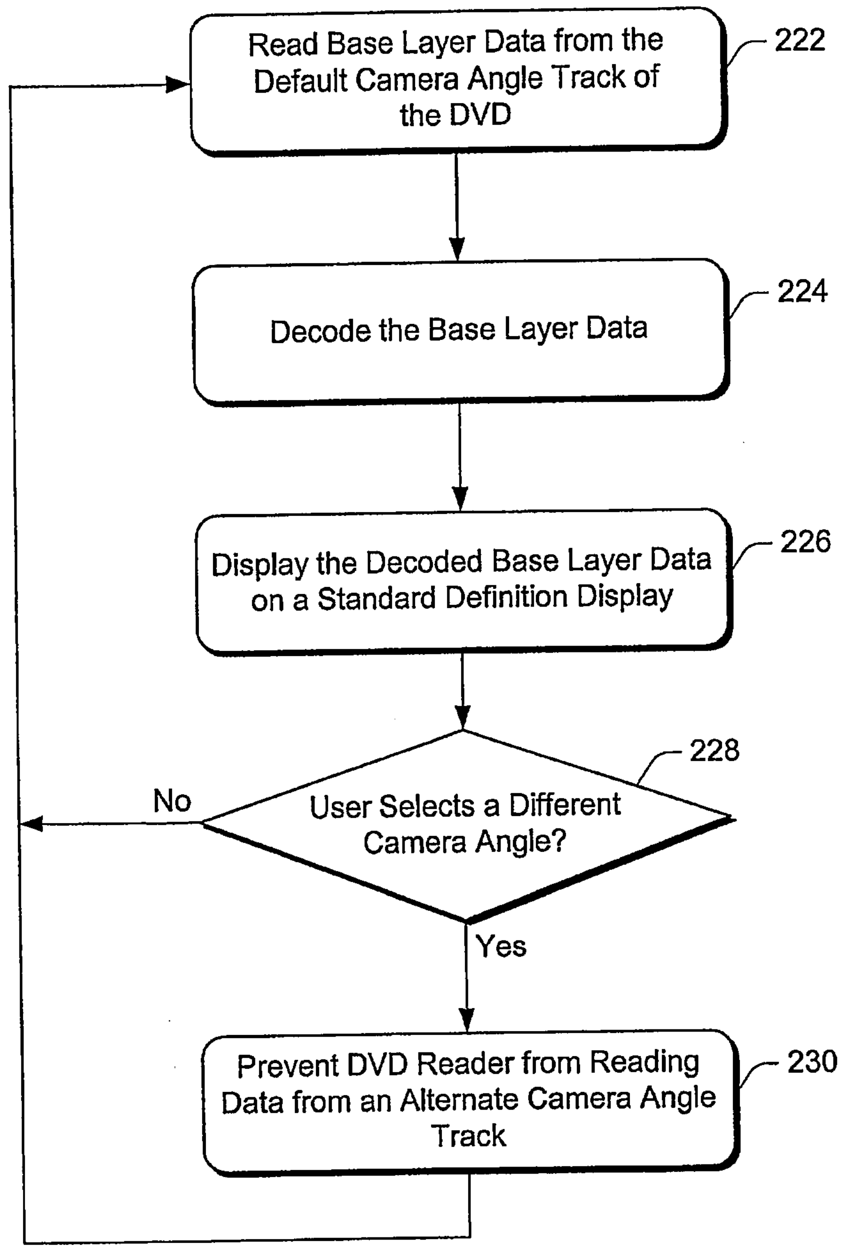

Read Base Layer Data from the Default Camera Angle Track of the DVD
222
Decode the Base Layer Data
224
Display the Decoded Base Layer Data on a Standard Definition Display
226
User Selects a Different Camera Angle?
228
No
Yes
Prevent DVD Reader from Reading Data from an Alternate Camera Angle Track
230

Fig. 4

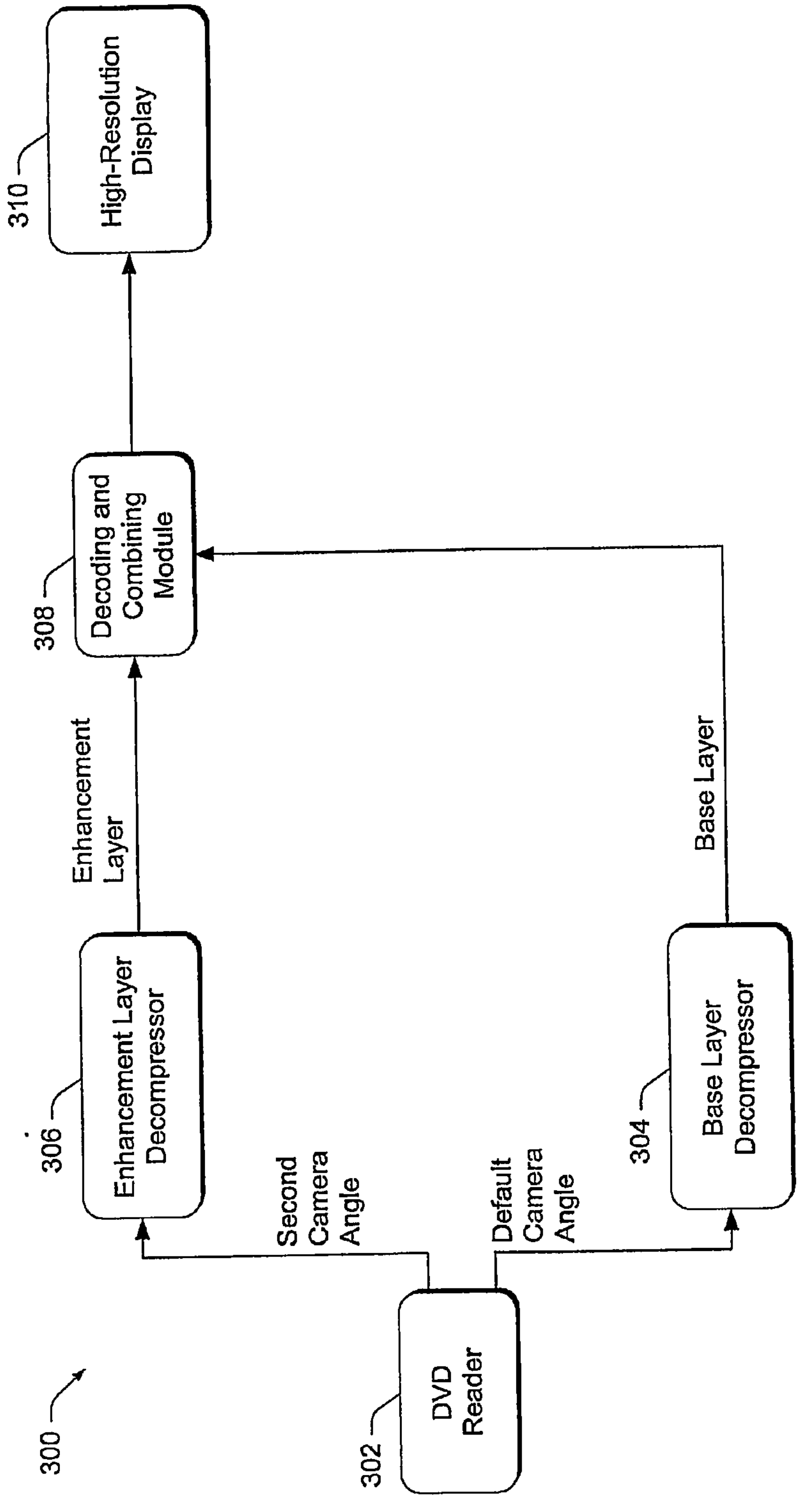

300
302
DVD Reader
Second Camera Angle
Default Camera Angle
306
Enhancement Layer Decompressor
304
Base Layer Decompressor
Enhancement Layer
Base Layer
308
Decoding and Combining Module
310
High-Resolution Display

Fig. 5

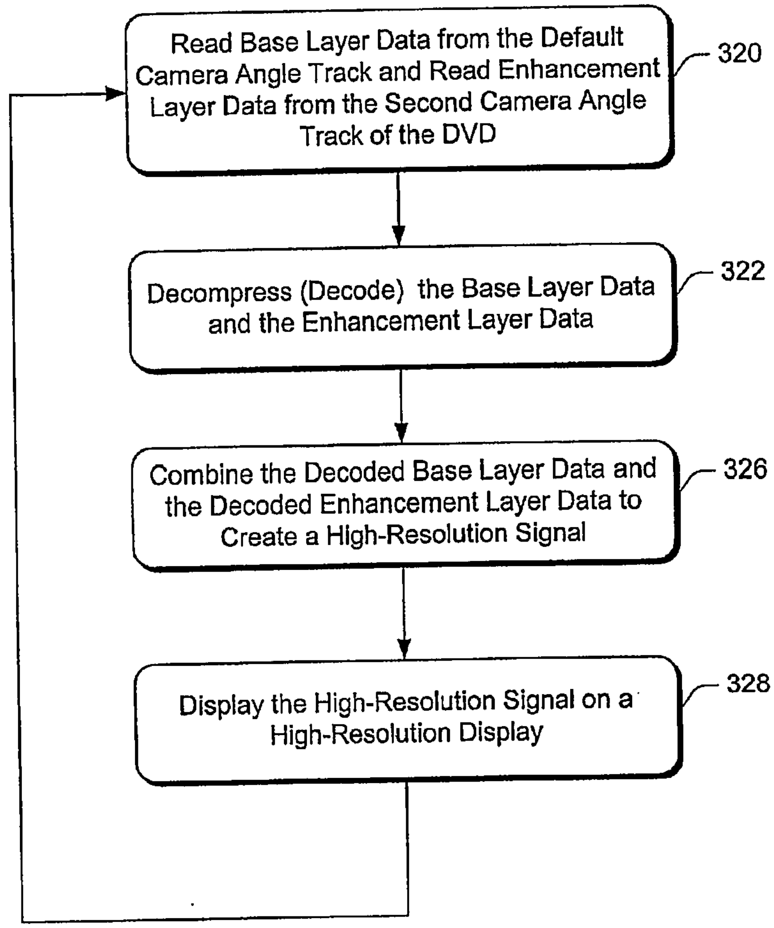
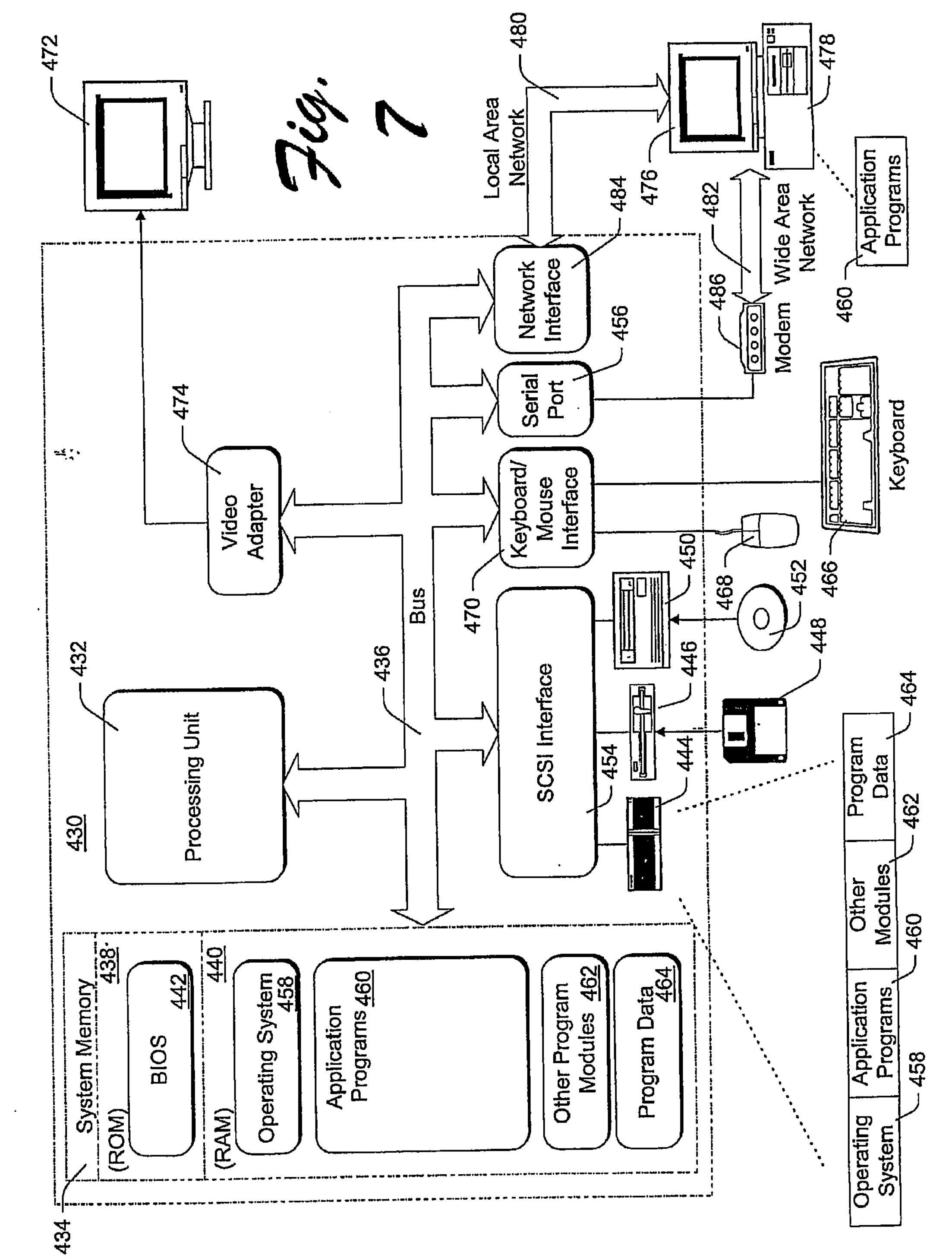

Read Base Layer Data from the Default Camera Angle Track and Read Enhancement Layer Data from the Second Camera Angle Track of the DVD
320
Decompress (Decode) the Base Layer Data and the Enhancement Layer Data
322
Combine the Decoded Base Layer Data and the Decoded Enhancement Layer Data to Create a High-Resolution Signal
326
Display the High-Resolution Signal on a High-Resolution Display
328

Fig. 6



430
Processing Unit
432
436
Bus
Video Adapter
474
472
SCSI Interface
454
Keyboard/ Mouse Interface
470
Serial Port
456
Network Interface
484
Local Area Network
480
476
478
482
Wide Area Network
Modem
486
Application Programs
460
Keyboard
466
468
450
452
446
448
444
Operating System
458
Application Programs
460
Other Modules
462
Program Data
464
System Memory
434
(ROM)
438
BIOS
442
(RAM)
440
Operating System 458
Application Programs 460
Other Program Modules 462
Program Data 464


Fig. 7

# LAYERED CODING OF IMAGE DATA USING SEPARATE DATA STORAGE TRACKS ON A STORAGE MEDIUM

## TECHNICAL FIELD

[0001] This invention relates to image processing systems. More particularly, the invention relates to systems that process images using a layered coding technique in which different tracks on a storage medium store different layers of data that can render either a standard definition or high resolution image while storing the data efficiently.

## BACKGROUND

[0002] Although a new high-definition television (HDTV) standard is emerging, most existing televisions and television receivers are low-resolution (i.e., standard definition televisions—SDTVs). Typically, the maximum resolution supported by a standard definition television is a horizontal resolution equivalent to 720 vertical lines by 480 interlaced horizontal scan lines with an effective resolution of approximately 350 lines of vertical resolution. The Advanced Television Systems Committee (ATSC) HDTV broadcast standard supports resolutions including 1280 ×720 lines per picture, which is approximately four times the number of pixels that can be resolved in a standard definition picture.

[0003] DVDs (Digital Video Discs or Digital Versatile Discs) are a popular medium for distributing video and audio/video programs, such as movies, musical concerts, and other video programs. The current DVD standard provides a maximum resolution of 720×480 for programs recorded on a DVD. Thus, the current DVD standard does not take advantage of the higher resolutions supported by HDTVs. Most DVDs are encoded from movie film or other storage media that supports the higher resolution of HDTVs. Therefore, the higher resolution version of the video program is typically available when the DVD is created, but the resolution is reduced to 720×480 (standard definition) when the DVD is manufactured.

[0004] As more HDTVs are manufactured and sold, more end users will desire DVDs having a higher resolution that matches the capability of their HDTV. However, to avoid obsoleting the large number of existing standard definition televisions and disc players, high-resolution DVD devices (e.g., high-resolution DVD players) will also need to support DVD programs recorded in the prior standard definition format.

[0005] One solution to this problem creates two different DVDs for each video program (e.g., one DVD that is encoded for standard definition devices and a different DVD encoded for high-resolution devices). This solution is undesirable because it requires the creation, distribution, and stocking of two different DVDs. Furthermore, until a large number of high-resolution DVD devices are sold in the marketplace, the cost of creating a small number of high-resolution DVDs may be too high.

[0006] Further, it would be undesirable to store two complete versions of a DVD title on the same disc (i.e., both a standard definition version and a high definition version). A high definition version would require the fall capacity of both physical layers of one side of a DVD, thus requiring an expensive dual-sided, dual-layer disc to also store the standard definition version of the title on the other side of the DVD. This is an inefficient and expensive solution because the standard definition data is stored twice on the same disc in two forms.

[0007] Therefore, a system is needed that allows both a standard definition version of a video program and a high-resolution version of the same program to be efficiently stored on a single DVD in a manner that allows the standard definition version to be compatible with existing equipment.

## SUMMARY

[0008] Layered coding, which separates a high-resolution image into a base layer and an enhancement layer, is described. A storage medium, such as a DVD, has at least two different data storage tracks (also referred to as data streams). One data storage track is used to store the base layer and the second data storage track stores the enhancement layer. A standard definition image is generated by decoding the base layer data. A high-resolution image is generated by decoding and combining both the base layer data and the enhancement layer data.

[0009] In one embodiment, an encoding system encodes a base layer representing a standard definition portion of a source image and encodes an enhancement layer representing a high-resolution portion of the source image. The base layer is stored on a first data storage track of a storage medium and the enhancement layer is stored on a second data storage track of the storage medium.

[0010] In another embodiment, the first data storage track is a default camera angle track and the second data storage track is a second camera angle track.

[0011] In a particular implementation of the system, the storage medium is a DVD.

[0012] Another embodiment provides a decoding system that decodes a base layer from a first data storage track of a storage medium and decodes an enhancement layer from a second data storage track of the storage medium.

[0013] In a described implementation, the base layer and the enhancement layer are decoded simultaneously.

[0014] A particular embodiment decodes the base layer from a default camera angle track and decodes the enhancement layer from a second camera angle track.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0015] **FIG. 1** illustrates a system that separates a high-resolution source image into a base layer and an enhancement layer, and stores the base layer and the enhancement layer in separate tracks on a storage medium.

[0016] **FIG. 2** is a flow diagram illustrating a procedure for encoding high-resolution source data into a base layer and an enhancement layer.

[0017] **FIG. 3** illustrates a standard definition DVD decoding system.

[0018] **FIG. 4** is a flow diagram illustrating a procedure for decoding a standard definition image from a DVD.

[0019] **FIG. 5** illustrates a high-resolution DVD decoding system.

[0020] **FIG. 6** is a flow diagram illustrating a procedure for decoding a high-resolution image from a DVD.

[0021] **FIG. 7** is a block diagram showing pertinent components of a computer in accordance with the invention.

## DETAILED DESCRIPTION

[0022] The system described herein provides a layered coding mechanism that separates a high-resolution source image into a base layer having a resolution appropriate for a typical standard definition television system and an enhancement layer which, when combined with the base layer, provides an image resolution appropriate for a high-resolution television system. The base layer is used by standard definition televisions that cannot utilize the higher resolution portions of the image contained in the enhancement layer. The enhancement layer contains the high-resolution portions of the source image, such as the sharp edges and the portions of the image with bright color and high contrast. High-definition devices, such as high-definition DVD players, connected to a high-resolution display device, such as an HDTV (high-definition television), use both the base layer and the enhancement layer to generate a high-resolution image on the television. Alternatively, a user of an HDTV may choose to view a particular video program in standard definition mode. In this situation, the HDTV uses only the base layer to generate a standard definition image on the television.

[0023] The base layer and the enhancement layer are stored in separate tracks on a storage medium such as a DVD (digital video disc or digital versatile disc). Tracks may be interleaved or multiplexed so that data from all tracks is read simultaneously, or tracks may be stored in separate physical locations on the storage medium. A conventional, standard definition DVD player reads and decodes only the base layer information from the DVD. An enhanced DVD player supports high-resolution televisions by reading and decoding both the base layer information and the enhancement layer information from the DVD. Thus, instead of requiring two different types of DVDs (one for standard definition DVD players and another for high-resolution DVD players), a single DVD can support both standard definition and high-resolution DVD players by reading and decoding the appropriate track(s) from the DVD. Thus, the single DVD supports both standard definition television systems as well as high-resolution television systems. As used herein, the term "DVD player" includes any device capable of reading data from a DVD disc or other medium and processing the data to generate video signals in accordance with the DVD format specification.

[0024] As used herein, the terms "television", "television system", and "television receiver" shall be understood to include any type of video display system, including a television, a television receiver, a video projector, a flat panel display, and related video display systems. Additionally, the term "video" includes any form of electronic imagery, such as film or digitized image sequences. Although particular examples are described herein that use a DVD as the storage medium, it will be understood that any type of storage medium having at least two data storage tracks can be used to implement the systems described herein.

[0025] Further, particular examples are described herein with reference to HDTV systems. However, it will be understood that the teachings provided herein can be applied to any type of high resolution or high definition video display system. The terms "high resolution" and "high definition", as used herein, are interchangeable.

[0026] The DVD video disc format permits the recording of multiple interleaved video tracks for uses such as allowing multiple selectable "video angles"**0** or "camera angles." For purposes of layered video resolution coding, the DVD video "video angles" or "camera angles" can be used as data tracks for video resolution layers.

[0027] **FIG. 1** illustrates a layered encoding system that separates a high-resolution source image into a base layer and an enhancement layer, and stores the base layer and the enhancement layer in separate tracks on a storage medium, such as a DVD. A layered encoding system may also be referred to as an image encoding system. A high-resolution source image **100** is captured using a video camera or other device capable of capturing an image. A series of successive source images are captured to generate a video program (e.g., a television program or a movie).

[0028] The high-resolution source image **100** is communicated to an enhancement layer generator **102** and a base layer generator **104**. The enhancement layer generator **102** generates an enhancement layer portion of the source image **100** and communicates the enhancement layer to a compressor **106**. The enhancement layer generator **102** generates the enhancement layer by comparing the base layer data (received from the base layer generator **104**) to the high-resolution source image data. For example, the enhancement layer generator **102** subtracts the base layer data from the high-resolution source image data, thereby leaving only the high-resolution portions of the image (i.e., the enhancement layer).

[0029] The base layer generator **104** generates a base layer portion of the source image **100** and communicates the base layer to a compressor **108**. The compressor **106** generates a compressed version of the enhancement layer data and the compressor **108** generates a compressed version of the base layer data. In a particular embodiment of the invention, compressor **108** compresses the base layer data using the MPEG-2 (moving picture experts group) compression algorithm. Similarly, compressor **106** may compress the enhancement layer using the MPEG-2 compression algorithm. However, compressor **106** is not required to use the same compression algorithm as compressor **108**. For example, compressor **106** may use a compression algorithm that utilizes three-dimensional wavelets to compress the enhancement layer information.

[0030] The compressed base layer is stored on a first data storage track **112** of storage medium **110**. A data storage track is a collection of multiple sectors on a storage medium that can be read in sequence in real time. For example, a data storage track on a DVD may be a continuous series of data elements stored in a generally circular pattern that are read as the DVD rotates. Alternatively, a data storage track on a DVD may store two interleaved streams of data, such as enhancement layer data interleaved with base layer data, in multiple sectors scattered over the DVD.

[0031] The compressed enhancement layer is stored on a second data storage track **114** of storage medium **110**. In this example, storage medium **110** is a DVD. The first and

second data storage tracks **112** and **114** may be located on the same physical layer of the DVD or may be located on different physical layers of the DVD (a DVD can have two sides with two physical layers on each side).

[0032] Compressors **106** and **108** compress the enhancement layer and base layer data to reduce the storage space required to store the data. If the enhancement layer and/or the base layer do not require compression (i.e., the storage medium **110** has sufficient storage space without compressing the data), then compressor **106** and/or **108** can be eliminated from the system shown in **FIG. 1**.

[0033] As mentioned above, the DVD format supports multiple camera angles (or video angles). A viewer of the program stored on a DVD may select the default camera angle or one of several alternate camera angles. Although DVD technology supports multiple camera angles, programs are not necessarily recorded using multiple camera angles. Due to the added cost of recording a video program using multiple camera angles, many programs do not utilize the DVD tracks provided for the alternate camera angles.

[0034] The first track **112** of the DVD is the track assigned to the default camera angle. The base layer data is stored on this default camera angle track since the base layer information is read by both standard definition and high-resolution systems. To maintain backward compatibility with existing DVD players, the base layer data is stored using the format defined in the DVD video specification. The enhancement layer data is stored on the second track **114**, which is assigned to an alternate camera angle. In this situation, the alternate camera angle track does not actually store data associated with an alternate camera angle, but instead stores data associated with the high-resolution portion of the source image. The enhancement layer contains special data sequences that allow a compatible high-definition DVD player to recognize that the camera angle track contains enhancement data. Although **FIG. 1** illustrates tracks **112** and **114** as two separate tracks, in one embodiment the two tracks are interleaved, or time division multiplexed, so that the two tracks can be read simultaneously. One or both of the interleaved tracks are read by demultiplexing the interleaved data packets.

[0035] **FIG. 2** is a flow diagram illustrating a procedure for encoding high-resolution source data into a base layer and an enhancement layer. The procedure illustrated in **FIG. 2** can be implemented, for example, using the layered encoding system described above with respect to **FIG. 1**. The encoding system receives a series of high-resolution source images (step **130**). Each source image is processed using the procedure of **FIG. 2**. The encoding system receives each high-resolution source image from a video camera or other image capture device (or video storage device). The high-resolution source image is communicated to an enhancement layer generator and a base layer generator (step **132**).

[0036] The flow diagram branches from step **132** into two parallel paths that are processed concurrently. Following the left path, the enhancement layer generator generates an enhancement layer (step **134**) using both the high-resolution source image and the base layer data generated by the base layer generator in step **140**. The enhancement layer data is then compressed (step **136**) and stored on the second track (i.e., the alternate camera angle track) of the DVD (step **138**).

[0037] Following the right path of **FIG. 2**, the base layer generator generates a base layer (step **140**). The base layer data is then compressed (step **142**) and stored on the first track (i.e., the default camera angle track) of the DVD (step **144**). At this point, the DVD contains both the compressed base layer data and the compressed enhancement layer data, stored on different tracks of the DVD. In an alternate embodiment, the base layer data and the enhancement layer data may be stored on an intermediate storage device, and later transferred onto a DVD. Furthermore, the base layer data and the enhancement layer data may be read by a device that manufactures the DVD by storing the appropriate data in the appropriate tracks.

[0038] **FIG. 3** illustrates a standard definition DVD decoding system **200**. A standard definition DVD reader **202** reads data from a default camera angle track of a DVD positioned in the DVD player. As mentioned above, the default camera angle track contains the base layer data. The DVD reader **202** may be located in a DVD player or other device coupled to a television for displaying the video program stored on the DVD. Alternatively, the DVD reader **202** may be located in a computer or other computing device for displaying the DVD's video program on a computer monitor or other display device.

[0039] A base layer decoder **204** decodes and decompresses the base layer information read from the DVD by reader **202**. The output of the base layer decoder **204** is the uncompressed base layer data that is understood by a standard definition display **206**. Standard definition display **206** displays the original sequence of images (in a standard definition mode). In the example of **FIG. 3**, base layer decoder **204** is shown as a separate device. In an alternate embodiment, the base layer decoder **204** may be incorporated into DVD reader **202** or standard definition display **206**. Alternatively, the base layer data stream generated by base layer decoder **204** is transmitted over a network (or transcoded to another format) for distribution to a remote device (such as a video display device or a storage device).

[0040] **FIG. 4** is a flow diagram illustrating a procedure for decoding a standard definition image from a DVD. A DVD reader reads the base layer data from the default camera angle track of the DVD (step **222**). The base layer data is then decoded (step **224**). The decoded base layer data is displayed on a standard definition display (step **226**), thereby recreating the original sequence of images. If the user attempts to select a different camera angle, the DVD reader is prevented from reading data from an alternate camera angle track (step **230**). In this procedure, only the standard definition image is being read from the DVD. Therefore, the DVD reader is limited to reading the base layer information contained in the default camera angle track. For example, the DVD reader may be incapable of interpreting the enhancement layer information contained in an alternate camera angle track. The procedure then continues reading base layer data from the default camera angle track of the DVD (step **222**).

[0041] The DVD reader is prevented from reading data from an alternate camera angle track, such as the track that contains the enhancement layer data, by disabling certain user operations (e.g., disabling the ability to change camera angles) in the DVD reader or control circuitry. This disabling of user operations is supported by the DVD specifi-

cation. Alternatively, each new segment of enhancement data stored on the second track may be interpreted by a standard definition reader as an instruction not to play that camera angle. Thus, if the user of the DVD reading device attempts to change to the second camera angle, the reader will read the instruction and either refuse to read the second camera angle or switch back to reading the default camera angle. Alternately, the second camera angle may contain data that causes a standard player to interpret it as blank video or as an empty angle. A high-resolution DVD reader, discussed below, understands that the second camera angle track contains enhancement layer data and processes the enhancement data accordingly.

[0042] **FIG. 5** illustrates a high-resolution DVD decoding system **300**, which is capable of reading and processing both the base layer data and the enhancement layer data to generate a high-resolution video program. A high-resolution DVD reader **302** reads compressed base layer data from a default camera angle track of a DVD positioned in the DVD player. Additionally, the DVD reader **302** reads compressed enhancement layer data from a second camera angle track of the DVD positioned in the DVD player. The DVD reader **302** ignores any instructions at the beginning of the enhancement layer data segments that would be interpreted by a standard definition reader as an instruction not to play that camera angle. The DVD reader **302** understands that the second camera angle track contains enhancement layer data, the instructions directed toward standard definition DVD readers are ignored.

[0043] Since the DVD reader **302** reads both the default camera angle track and the second camera angle track, the DVD reader spins the DVD at twice the "standard rotational speed", or faster. In a particular embodiment, the standard rotational speed allows the DVD reader **302** to read one camera angle at approximately 8 Mbps (megabits per second). If the DVD reader spins the DVD at twice the standard rotational speed, then the DVD reader **302** can read two different camera angles simultaneously at approximately 16 Mbps.

[0044] A base layer decompressor **304** decompresses the compressed base layer data read from the DVD by reader **302**. Similarly, an enhancement layer decompressor **306** decompresses the compressed enhancement layer data read from the DVD by reader **302**. The outputs of decompressor **304** and decompressor **306** are coupled to a decoding and combining module **308**, which decodes and combines the base layer data with the enhancement layer data to generate a high-resolution signal that is provided to and understood by a high-resolution display **310**. High-resolution display **310** displays the original sequence of images in a high-resolution mode. In the example of **FIG. 5**, decompressors **304** and **306**, and the decoding and combining module **308** are shown as separate devices. However, any one or more of the devices can be incorporated into DVD reader **302** and/or high-resolution display **310**. In another embodiment, the data output from decoding and combining module **308** is transmitted over a network (such as the Internet) or other communication medium to a remote device (such as a video display device or a storage device).

[0045] Alternatively, the decoding and combining module **308** may generate an encoded high-definition MPEG-2 stream (or transcode to another encoded format), or could provide the decoded video to a distribution device (not shown) for transmission to remote devices. Although not shown in **FIG. 5**, the output of base layer decompressor **304** may also be coupled to a standard definition display device for displaying the video content at a standard resolution.

[0046] **FIG. 6** is a flow diagram illustrating a procedure for decoding a high-resolution image from a DVD. A DVD reader reads base layer data from the default camera angle track and reads enhancement layer data from the second camera angle track of the DVD (step **320**). Next, the procedure decompresses (decodes) the base layer data and the enhancement layer data (step **322**). The decoded base layer data and the decoded enhancement layer data are combined to create a high-resolution signal (step **326**). Finally, the high-resolution signal is displayed on a high-resolution display (step **328**), which recreates the original sequence of images. The procedure then returns to step **320** to continues reading base layer data from the default camera angle track and reading enhancement layer data from the second camera angle track of the DVD.

[0047] **FIG. 7** is a block diagram showing pertinent components of a computer **430** that can be used with the present invention. A computer such as that shown in **FIG. 7** can be used, for example, to perform various procedures necessary to encode or decode images, to store image data for later retrieval, read data from a DVD, or to display images on a display device coupled to the computer.

[0048] Computer **430** includes one or more processors or processing units **432**, a system memory **434**, and a bus **436** that couples various system components including the system memory **434** to processors **432**. The bus **436** represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory **434** includes read only memory (ROM) **438** and random access memory (RAM) **440**. A basic input/output system (BIOS) **442**, containing the basic routines that help to transfer information between elements within computer **430**, such as during start-up, is stored in ROM **438**.

[0049] Computer **430** further includes a hard disk drive **444** for reading from and writing to a hard disk (not shown), a magnetic disk drive **446** for reading from and writing to a removable magnetic disk **448**, and an optical disk drive **450** for reading from or writing to a removable optical disk **452** such as a CD ROM, DVD or other optical media. The hard disk drive **444**, magnetic disk drive **446**, and optical disk drive **450** are connected to the bus **436** by an SCSI interface **454** or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for computer **430**. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk **448** and a removable optical disk **452**, it should be appreciated by those skilled in the art that other types of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment.

[0050] A number of program modules may be stored on the hard disk **444**, magnetic disk **448**, optical disk **452**, ROM

438, or RAM 440, including an operating system 458, one or more application programs 460, other program modules 462, and program data 464. A user may enter commands and information into computer 430 through input devices such as a keyboard 466 and a pointing device 468. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 432 through an interface 470 that is coupled to the bus 436. A monitor 472 or other type of display device is also connected to the bus 436 via an interface, such as a video adapter 474. Video adapter 474 can be, for example, a DVD decoder combined with a SVGA display adapter to provide a SVGA signal to a SVGA monitor. Video adapter 474 can be implemented in hardware or software. In addition to the monitor 472, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.

[0051] Computer 430 commonly operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 476. The remote computer 476 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 430, although only a memory storage device 478 has been illustrated in FIG. 7. The logical connections depicted in FIG. 7 include a local area network (LAN) 480 and a wide area network (WAN) 482. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

[0052] When used in a LAN networking environment, computer 430 is connected to the local network 480 through a network interface or adapter 484. When used in a WAN networking environment, computer 430 typically includes a modem 486 or other means for establishing communications over the wide area network 482, such as the Internet. The modem 486, which may be internal or external, is connected to the bus 436 via a serial port interface 456. In a networked environment, program modules depicted relative to the personal computer 430, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0053] Generally, the data processors of computer 430 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described herein in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described herein.

[0054] For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

[0055] Alternatively, the invention can be implemented in hardware, software, or a combination of hardware, software, and/or firmware. For example, one or more application specific integrated circuits (ASICs) could be programmed to carry out the invention.

[0056] Although an exemplary system has been described using a two-layer coding system (i.e., base layer and enhancement layer), alternate embodiments may encode a source signal into any number of layers, each of which is stored as a separate track on a DVD.

[0057] Thus, a system has been described that provides a layered coding system that separates a high-resolution source image into a base layer and an enhancement layer, each of which are stored on a separate track of the storage medium. In a particular application, the base layer is stored on a default camera angle track and the enhancement layer is stored on a second camera angle track of the storage medium.

[0058] Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

1. A method of encoding a source image, the method comprising:

encoding a base layer representing a standard definition portion of the source image, wherein the base layer is stored on a first data storage track of a storage medium; and

encoding an enhancement layer representing a high-resolution portion of the source image, wherein the enhancement layer is stored on a second data storage track of the storage medium.

2. A method as recited in claim 1 wherein the storage medium is a DVD.

3. A method as recited in claim I wherein the first data storage track is a default camera angle track.

4. A method as recited in claim 1 wherein the second data storage track is a second camera angle track.

5. A method as recited in claim 1 wherein the first data storage track is interleaved with the second data storage track.

6. A method as recited in claim 1 wherein encoding an enhancement layer includes identifying the second track as being used to store enhancement layer data.

7. A method as recited in claim 1 wherein the base layer is encoded on a first physical layer of the storage medium and the enhancement layer is encoded on a second physical layer of the storage medium.

8. A method as recited in claim 1 wherein enhancement layer data is formatted such that a standard definition device is prevented from reading the enhancement layer data.

9. One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 1.

10. A method comprising:

decoding a base layer from a first data storage track of a storage medium, wherein the base layer represents a standard definition portion of an encoded image; and

decoding an enhancement layer from a second data storage track of the storage medium if the data stored on the second data storage track is identified as enhancement layer data, wherein the enhancement layer data represents a high-resolution portion of the encoded image.

11. A method as recited in claim 10 wherein decoding a base layer is performed simultaneously with decoding an enhancement layer.

12. A method as recited in claim 10 wherein the storage medium is a DVD.

13. A method as recited in claim 10 wherein the first data storage track is a default camera angle track.

14. A method as recited in claim 10 wherein the second data storage track is a second camera angle track.

15. A method as recited in claim 10 further including communicating the base layer to a standard definition television.

16. A method as recited in claim 10 further including combining the base layer and the enhancement layer to generate a high-resolution image.

17. A method as recited in claim 10 wherein the method is executed by a television.

18. A method as recited in claim 10 wherein the method is executed by a device capable of reading a DVD.

19. One or more computer-readable memories containing a computer program that is executable by a processor to perform the method recited in claim 10.

20. A computer-readable medium comprising:

a first data storage track to store a first layer of a video program, wherein the first layer represents a low-resolution portion of the video program and the first data storage track is a default camera angle track; and

a second data storage track to store a second layer of a video program, wherein the second layer represents a high-resolution portion of the video program and the second data storage track is a second camera angle track.

21. A computer-readable medium as recited in claim 20 wherein the computer-readable medium is a DVD.

22. A computer-readable medium as recited in claim 20 wherein the first data storage track is interleaved with the second data storage track.

23. A computer-readable medium as recited in claim 20 wherein the data is formatted such that a low-resolution device will not read the second layer data.

24. A DVD comprising:

a first camera angle track to store a base layer of a video program, wherein the base layer represents a standard definition portion of the video program; and

a second camera angle track to store an enhancement layer of a video program, wherein the enhancement layer represents a high-resolution portion of the video program.

25. A DVD as recited in claim 24 wherein the data is formatted such that a standard definition device will not read the enhancement layer data.

26. A DVD as recited in claim 24 wherein the first camera angle track is interleaved with the second camera angle track.

27. An apparatus comprising:

a reading device to read base layer data from a first track of a storage medium and to read enhancement layer data from a second track of the storage medium;

a decoder coupled to the reading device to decode any encoded data read from the first and second tracks of the storage medium; and

a combining module coupled to the decoder and the reading device to combine data read from the first track and data read from the second track into video program data.

28. An apparatus as recited in claim 27 wherein the apparatus is a device capable of reading a DVD.

29. An apparatus as recited in claim 27 wherein the apparatus is a computer.

30. An apparatus as recited in claim 27 wherein the storage medium is a DVD.

31. An apparatus having a reader capable of reading base layer data from a first data storage track of a storage medium and reading enhancement layer data from a second data storage track of the storage medium, the apparatus comprising a combining module coupled to the reader to combine data read from the first data storage track and data read from the second data storage track into video program data.

32. An apparatus as recited in claim 31 wherein the first data storage track is a default camera angle track.

33. An apparatus as recited in claim 31 wherein the second data storage track is a second camera angle track.

34. An apparatus as recited in claim 31 wherein the base layer data represents a standard resolution portion of a source image and the enhancement layer data represents a high-resolution portion of the source image.

35. An apparatus as recited in claim 31 wherein the combining module generates a high-resolution image.

36. An apparatus comprising:

a base layer generator to generate base layer data representing a standard definition portion of a source image, wherein the base layer data is located on a default camera angle track of a storage medium; and

an enhancement layer generator to generate enhancement layer data representing a high-resolution portion of the source image, wherein the enhancement layer data is located on a second camera angle track of the storage medium.

37. An apparatus as recited in claim 36 further including a first compressor coupled to the base layer generator to compress the base layer data.

38. An apparatus as recited in claim 36 further including a second compressor coupled to the enhancement layer generator to compress the enhancement layer data before the enhancement layer data.

39. One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:

generate a first layer representing a low-resolution portion of a source image, wherein the first layer is stored on a first data track of a storage medium; and

generate a second layer representing a high-resolution portion of the source image, wherein the second layer is stored on a second data track of the storage medium.

**40**. One or more computer-readable media as recited in claim 39 wherein the storage medium is a DVD.

**41**. One or more computer-readable media as recited in claim 39 wherein the first data track is interleaved with the second data track.

**42**. One or more computer-readable media as recited in claim 39 wherein the second layer is formatted such that a low-resolution device will ignore the second layer.

**43**. One or more computer-readable media having stored thereon a computer program that, when executed by one or more processors, causes the one or more processors to:

decode a base layer from a first camera angle track of a storage medium, wherein the base layer represents a standard definition portion of an encoded image; and

decode an enhancement layer from a second camera angle track of the storage medium, wherein the enhancement layer represents a high-resolution portion of the encoded image.

**44**. One or more computer-readable media as recited in claim 43 wherein the base layer and the enhancement layer are decoded simultaneously.

**45**. One or more computer-readable media as recited in claim 43 wherein the storage medium is a DVD.

**46**. One or more computer-readable media as recited in claim 43 wherein the first camera angle track is a default camera angle track.

**47**. One or more computer-readable media as recited in claim 43 wherein the one or more processors further communicate the base layer to a standard definition display device.

**48**. One or more computer-readable media as recited in claim 43 wherein the one or more processors further combine the base layer and the enhancement layer to generate a high-resolution image.

* * * * *

US 20040022318A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0022318 A1

Garrido et al. (43) Pub. Date: Feb. 5, 2004

(54) VIDEO INTERPOLATION CODING

(76) Inventors: **Diego Garrido**, Newtown, PA (US); **Richard Webb**, Belmont, CA (US); **Simon Butler**, San Rafael, CA (US); **Chad Fogg**, Seattle, WA (US)

Correspondence Address:
**BRINKS HOFER GILSON & LIONE**
**P.O. BOX 10395**
**CHICAGO, IL 60611 (US)**

(21) Appl. No.: **10/447,213**

(22) Filed: **May 28, 2003**

**Related U.S. Application Data**

(60) Provisional application No. 60/384,047, filed on May 29, 2002.

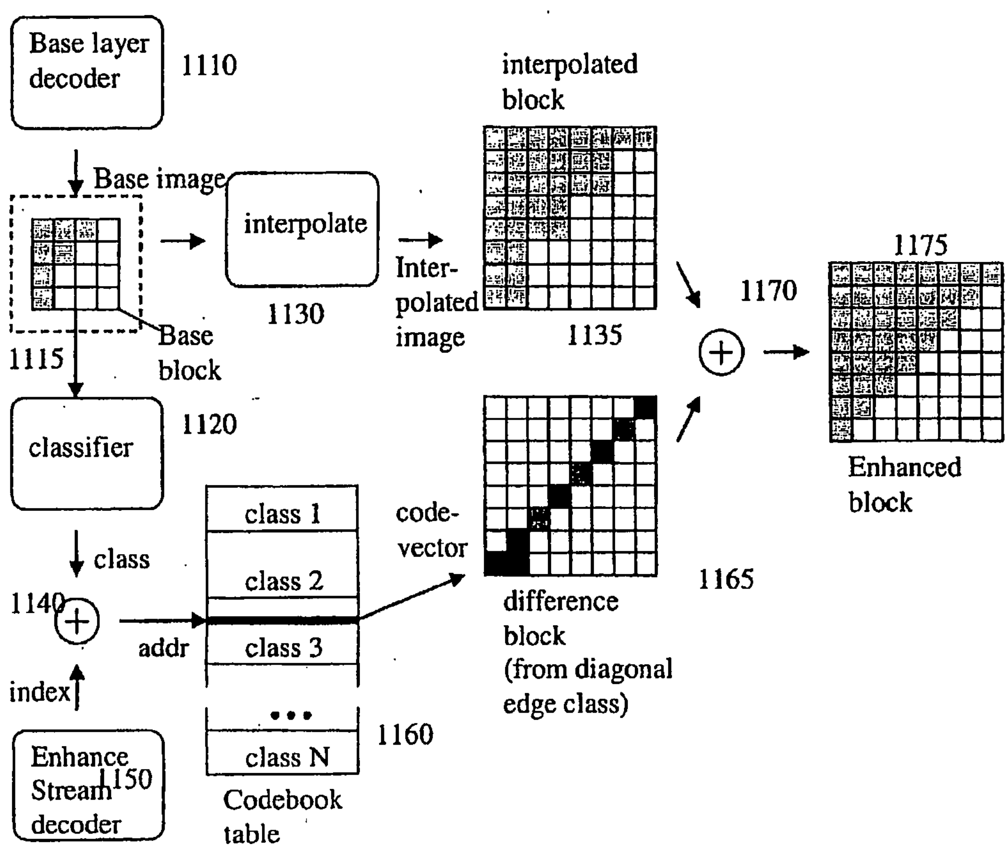
**Publication Classification**

(51) **Int. Cl.**[7] .................................................... **H04N 7/12**

(52) **U.S. Cl.** ................................ **375/240.11**; 375/240.29

(57) **ABSTRACT**

A method of enhancing picture quality of a video signal is described. The method comprising steps of receiving base images of pictures having a first definition from a base layer decoder; coding the differences between the base images of pictures and pictures having a second definition using vector quantization; creating a database of codebooks based upon the differences; and generating enhanced images based upon the base images and enhancement stream data. A circuit for enhancing picture quality of a video signal is also described. The circuit comprises a base layer decoder generating a base image of a standard definition picture; an interpolator coupled to the base layer decoder and generating an interpolated block; a classifier coupled to the base layer decoder and generating a class number; and a summing circuit coupled to the interpolator and the classifier. The summing circuit preferably adds the interpolated block and a difference block.



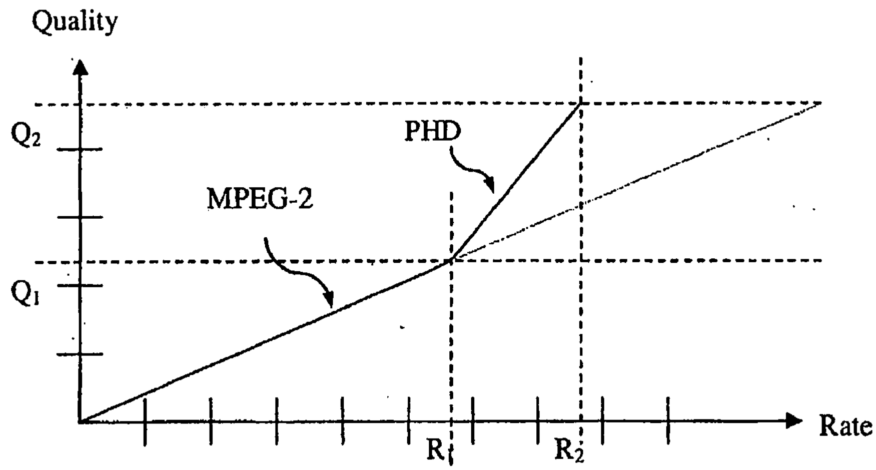



**Figure 1a-- Rate-distortion of MPEG-2 / PHD layered combination**

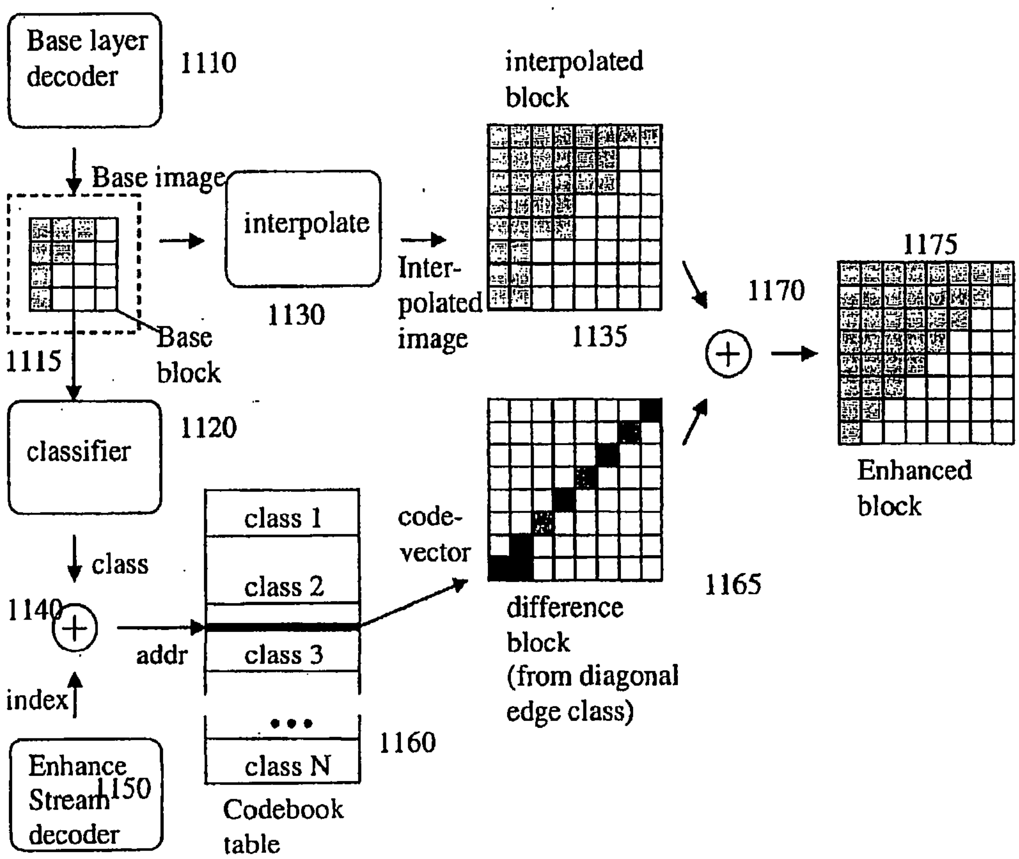



**Figure 1b**

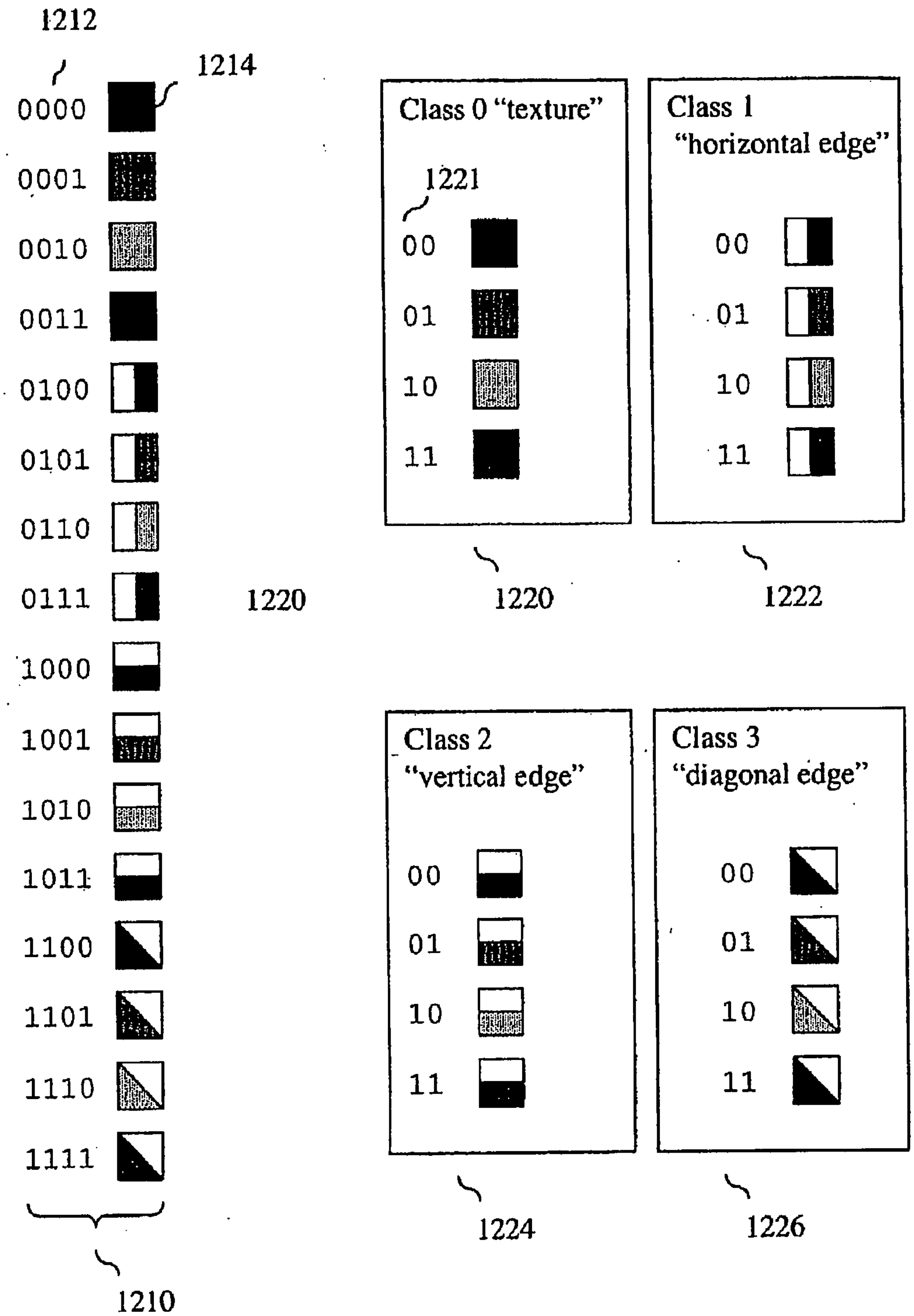



**Figure 1c -- Before and after classification patitioning**

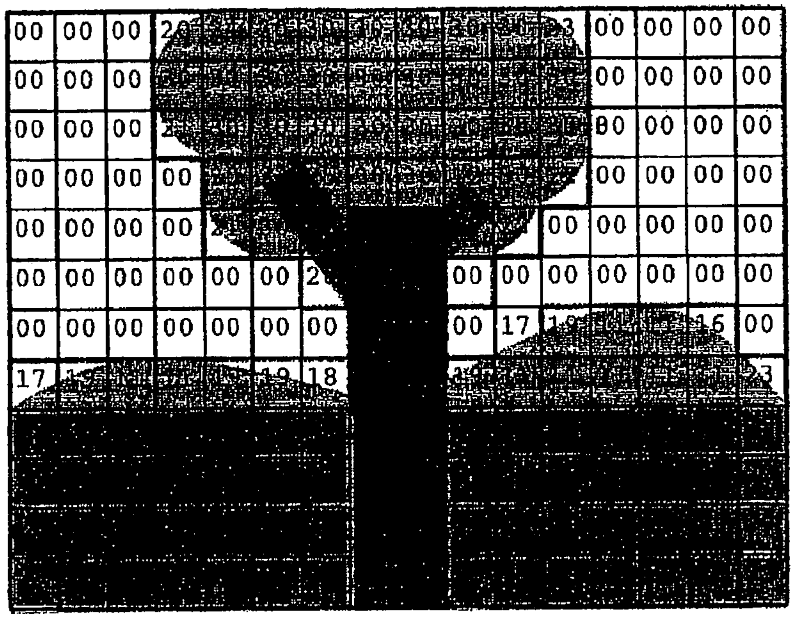



Figure 1d -- example classified image

By index:

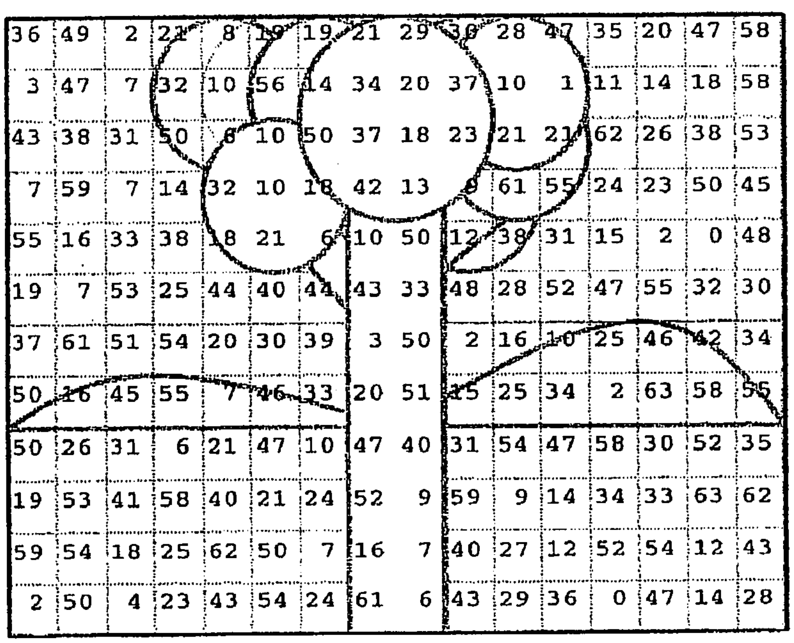



Figure 1e -- example indices for image

PRIOR ART SCALABLE SCHEMES

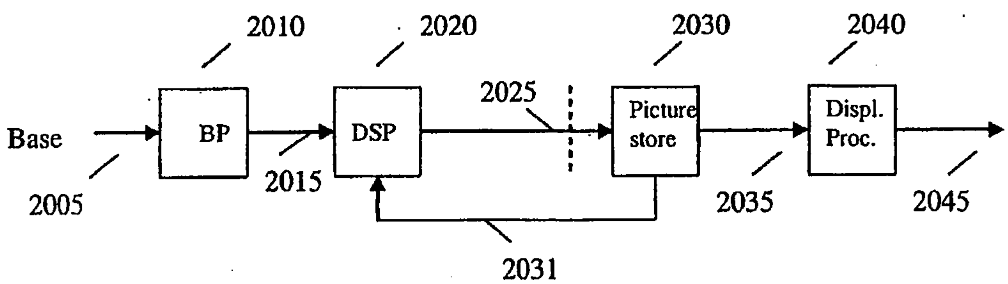



Fig.2a – One non scalable stream

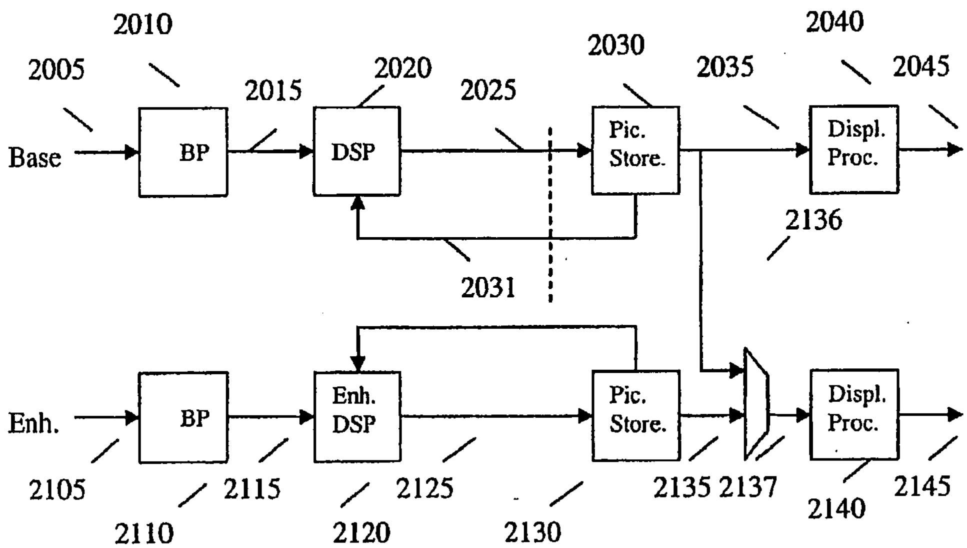



Fig. 2b – Two non-scalable streams, independently decodable ("simulcast")

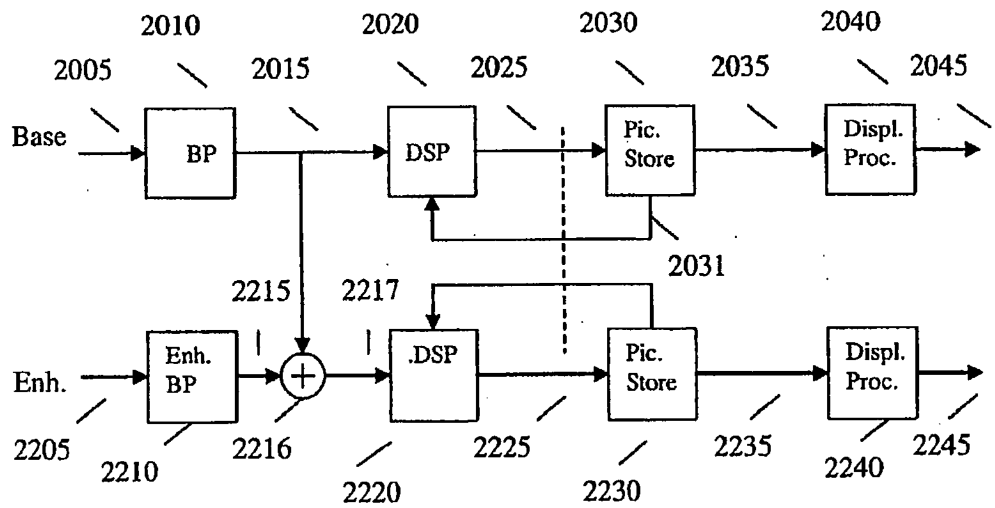



Fig 2c – Layering by token (SNR scalability, data partitioning)

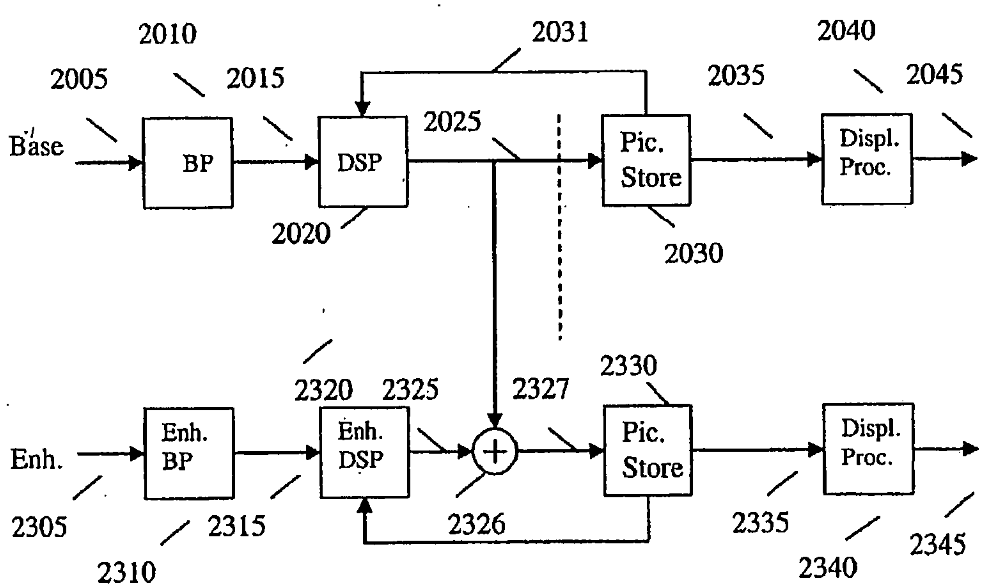



Fig. 2d – Layering by sample (Spatial scalability)

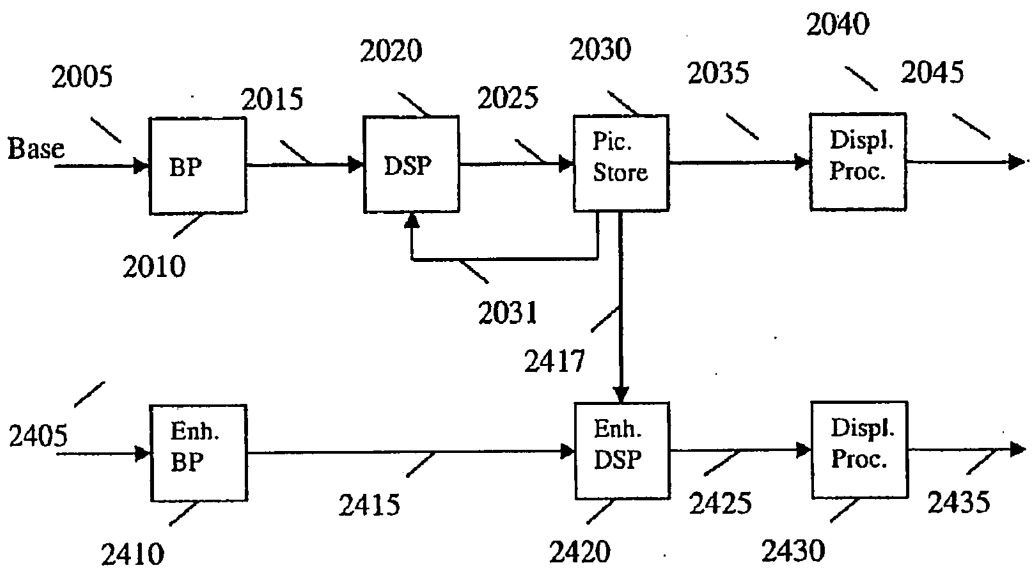



**Figure 2e – layering by frame (temporal scalability)**

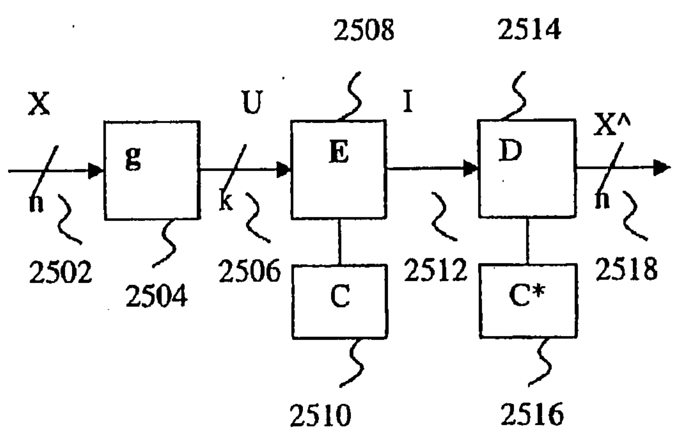



**Figure 2f Gersho NLIVQ '90 (Fig.2)**

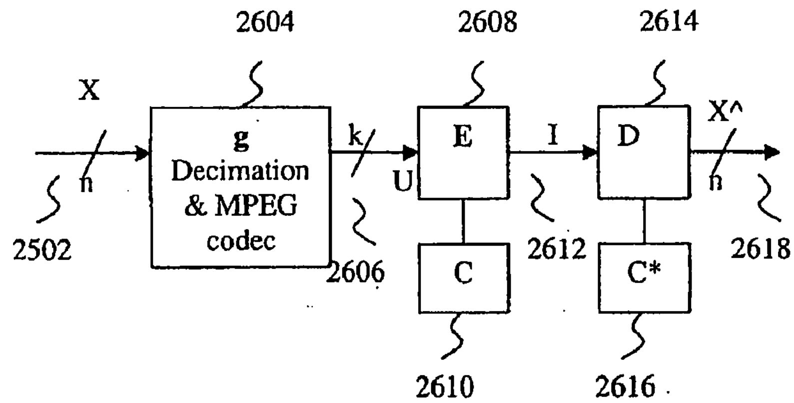



**Figure 2g Gersho NLIVQ '90 adopted to interpolation of MPEG coded video**

*g = downsampling + quantization via MPEG.*

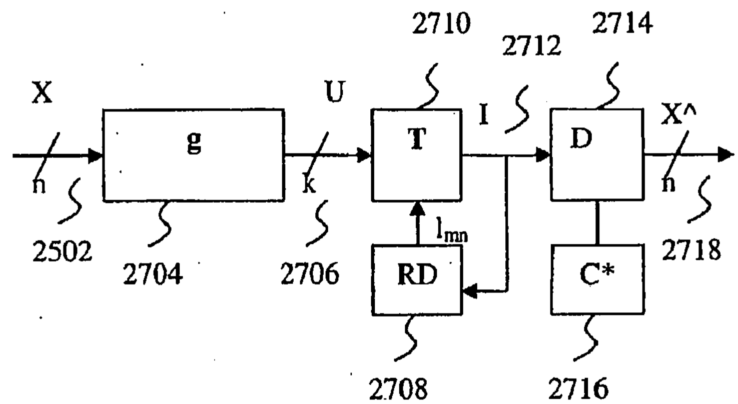



**Figure 2h Sheppard NLIVQ '96**

*T = encoder: DCT transform + quantization.*
*RD = quantizer design*

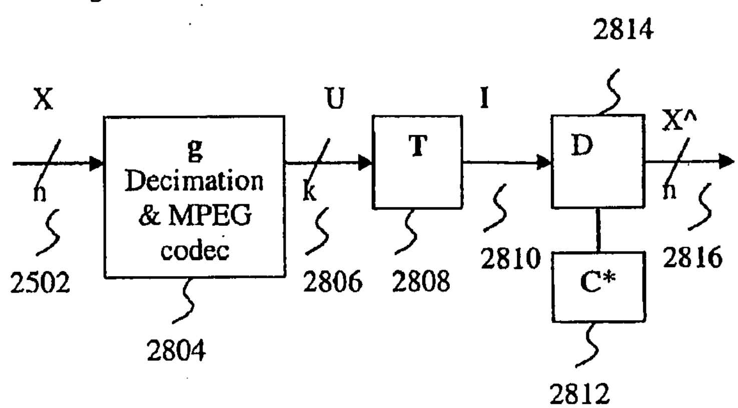



**Figure 2i Sheppard NLIVQ '96 adopted to interpolation of MPEG coded video**

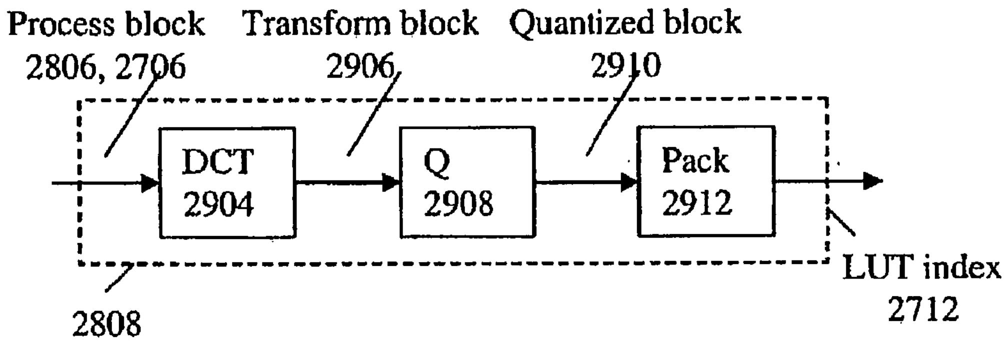



**Figure 2i -- Index generation steps, Sheppard et al '96**

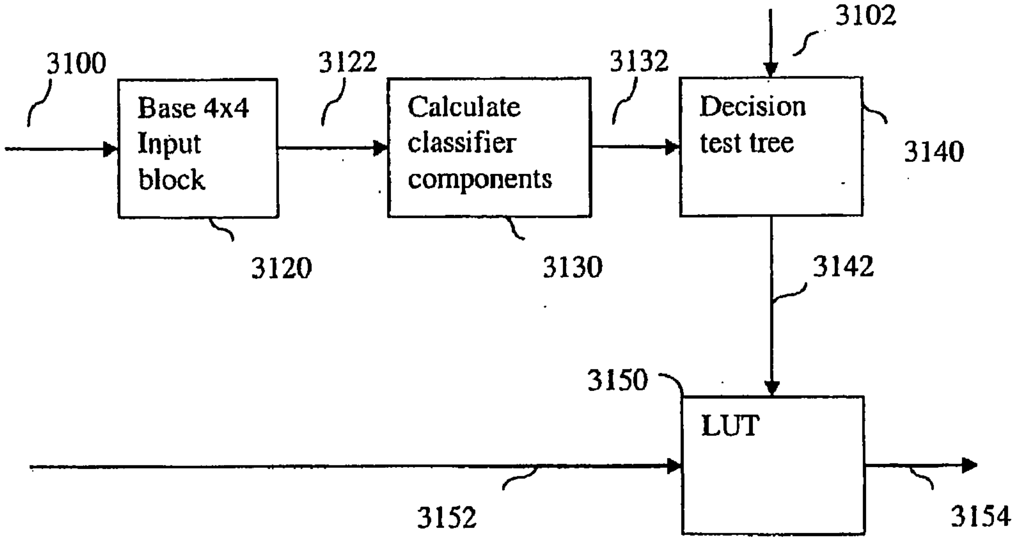



Fig. 3b

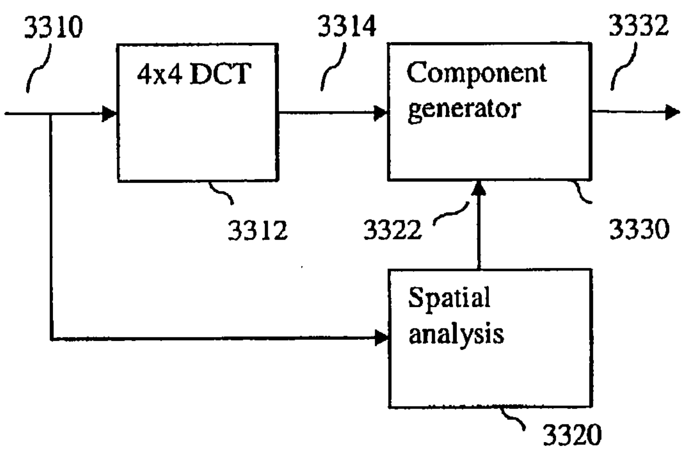



Figure 3d Calculate classifier components

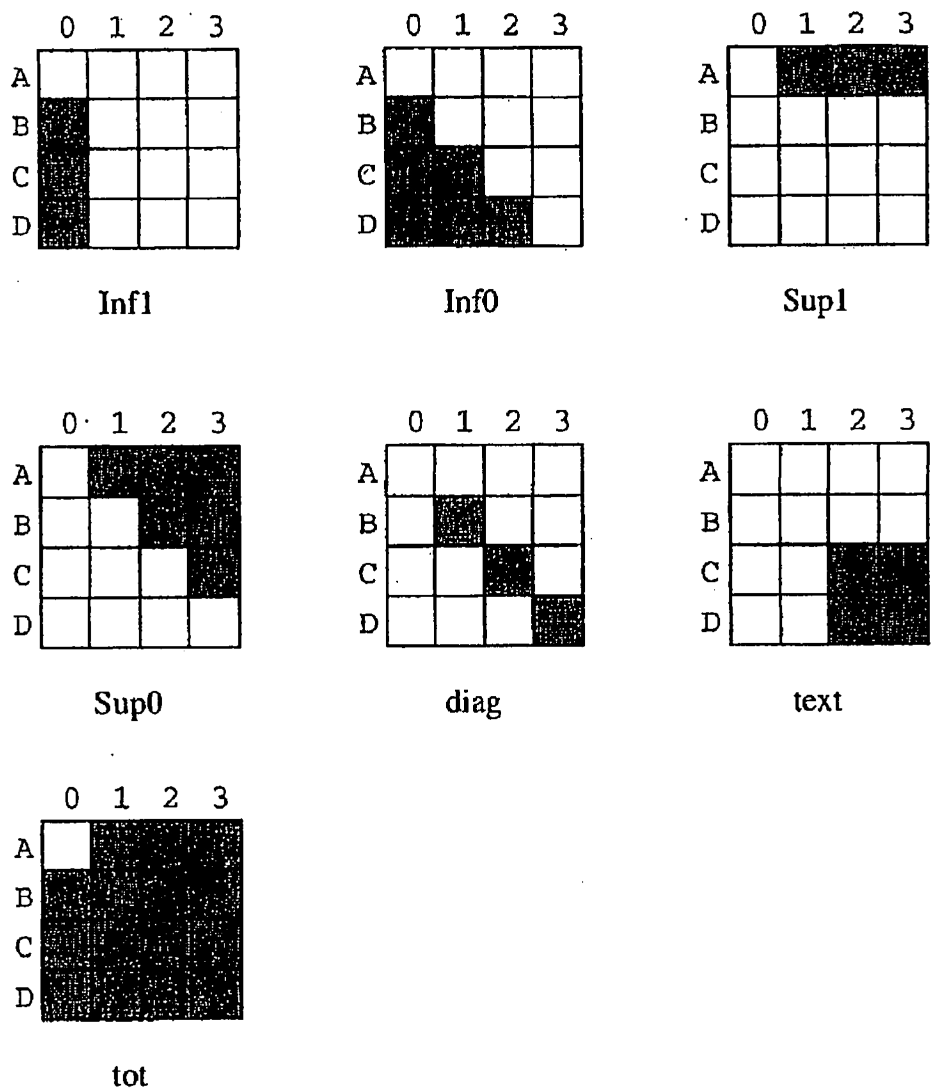



Figure 3e

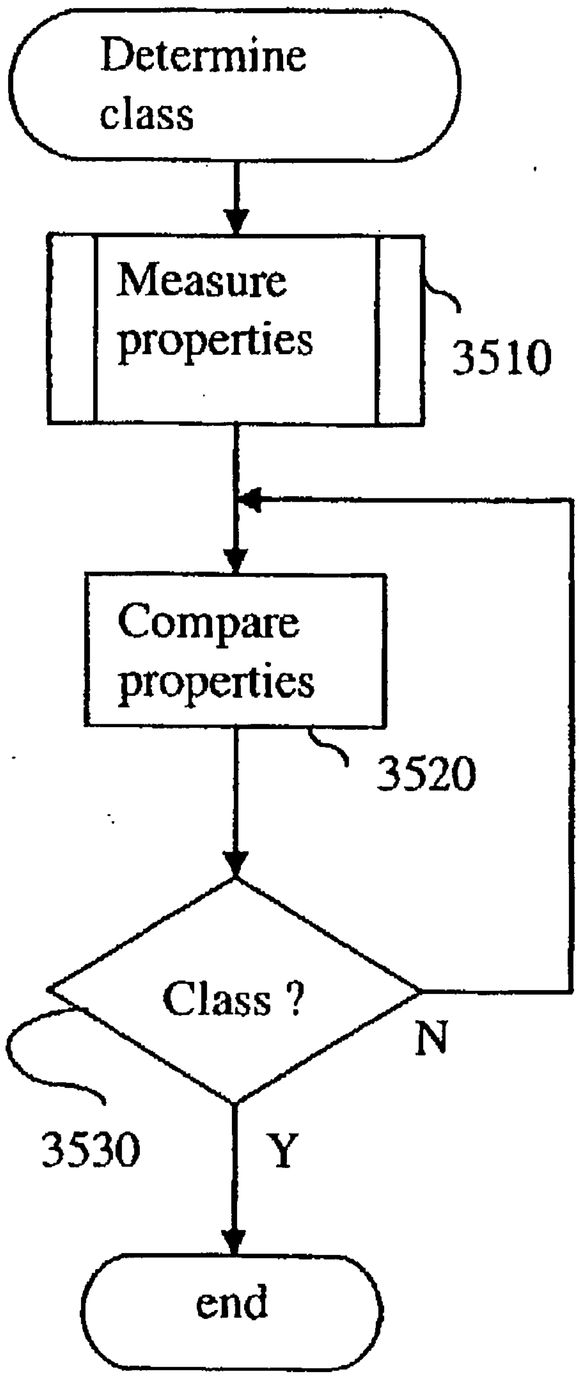

Determine class
Measure properties
3510
Compare properties
3520
Class ?
N
3530
Y
end

Figure 3f

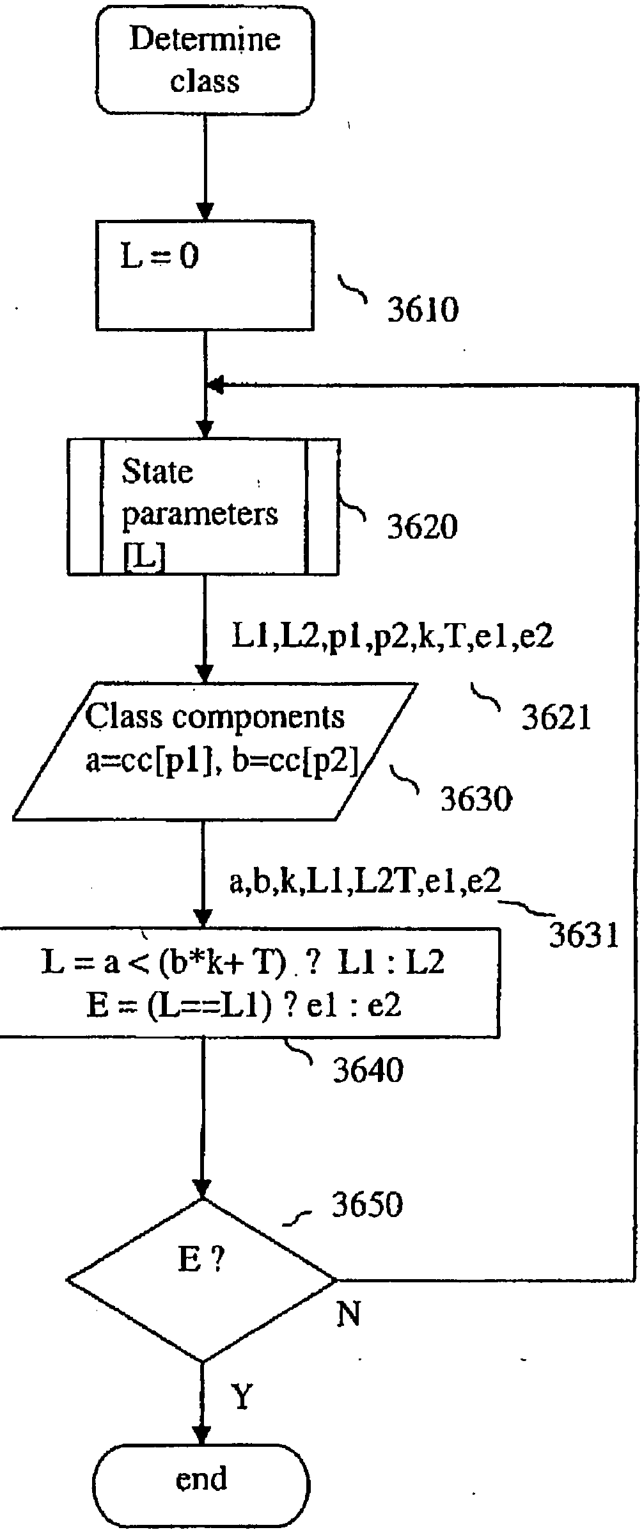

Determine class
L = 0
3610
State parameters [L]
3620
L1,L2,p1,p2,k,T,e1,e2
3621
Class components a=cc[p1], b=cc[p2]
3630
a,b,k,L1,L2T,e1,e2
3631
L = a < (b*k+ T) ? L1 : L2
E = (L==L1) ? e1 : e2
3640
3650
E ?
N
Y
end

Figure 3g

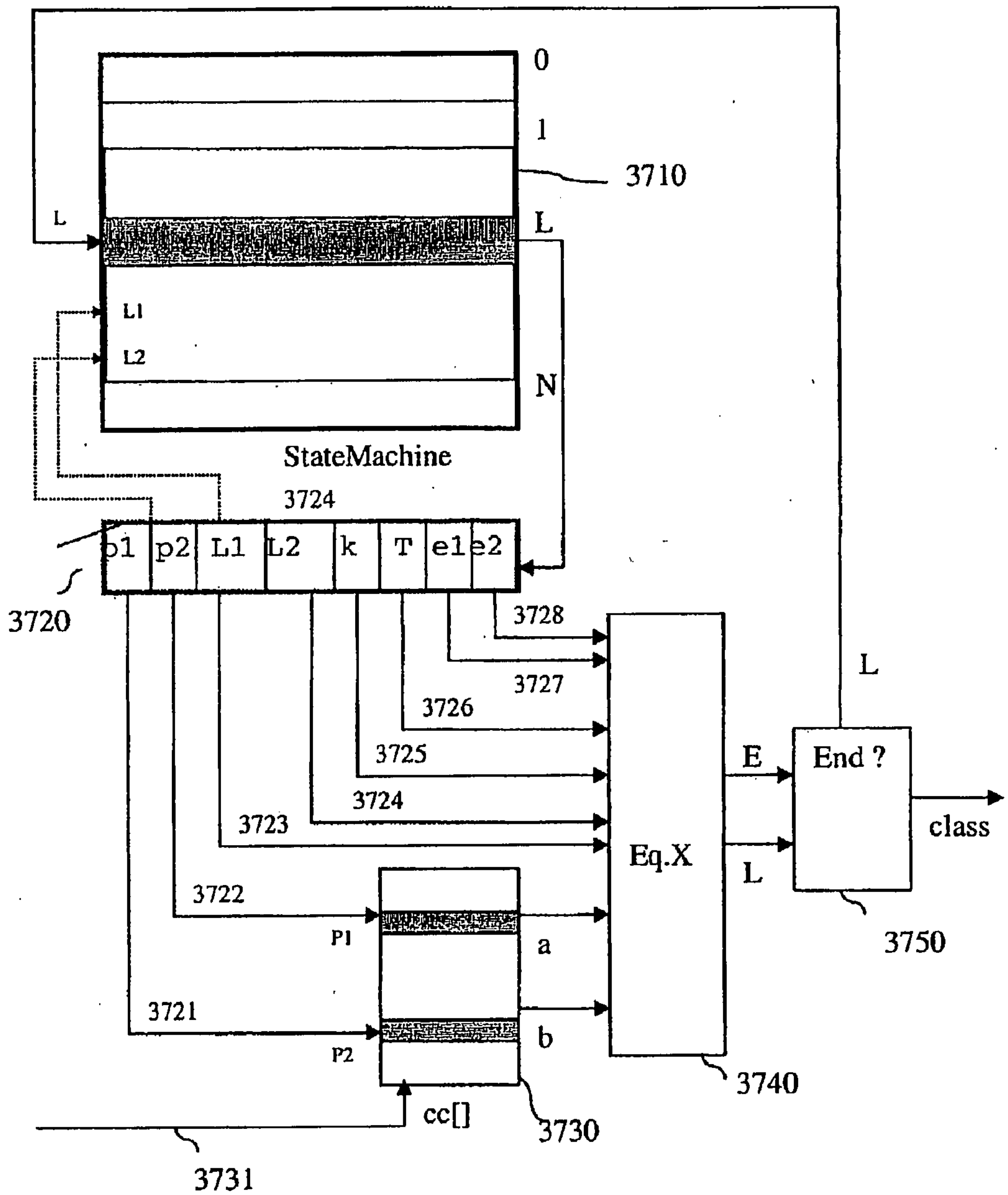
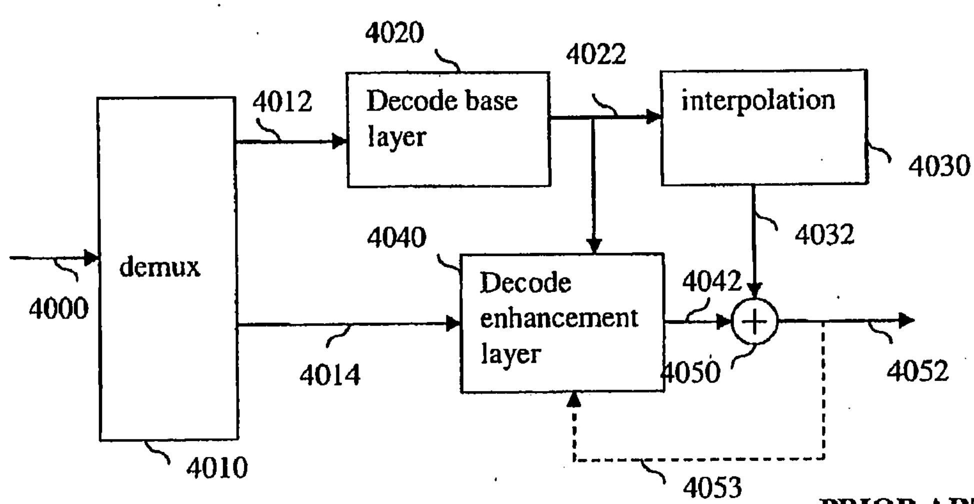



Figure 3b State machine

Fig. 4, decoder




PRIOR ART

Fig. 4a

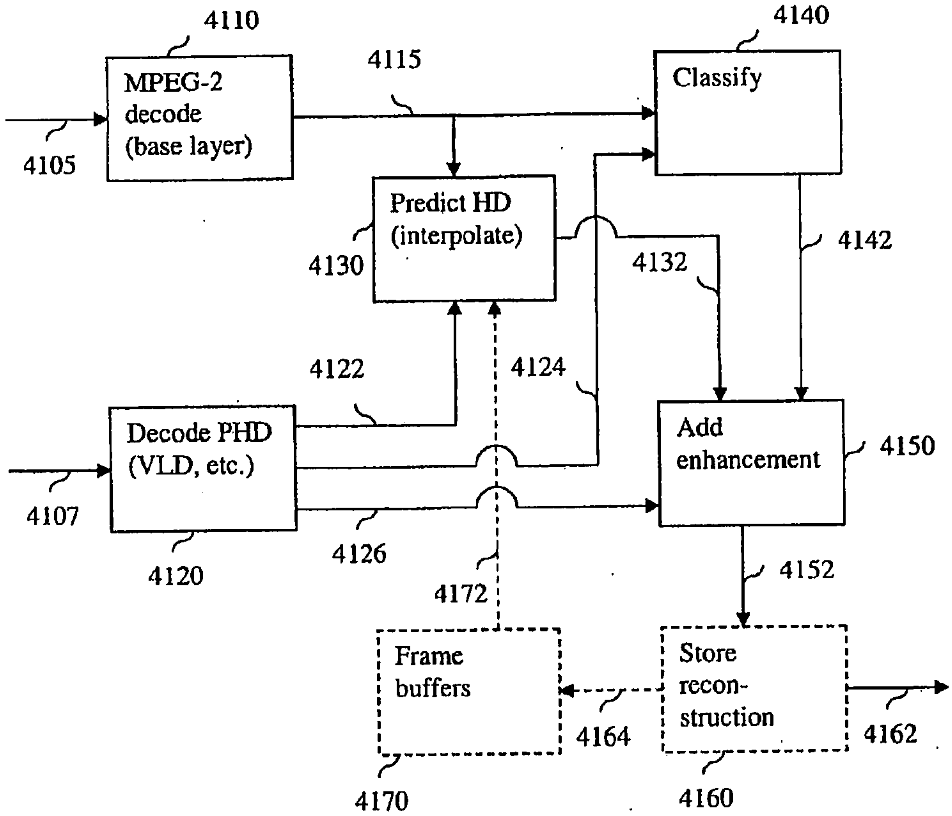
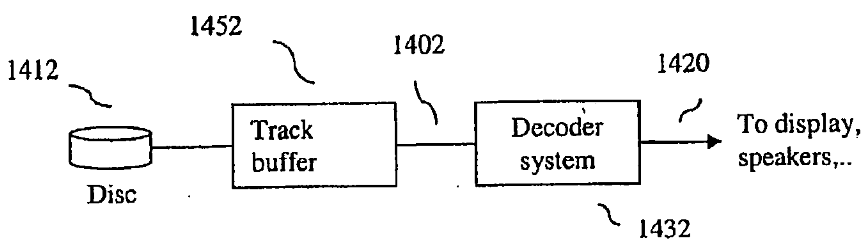
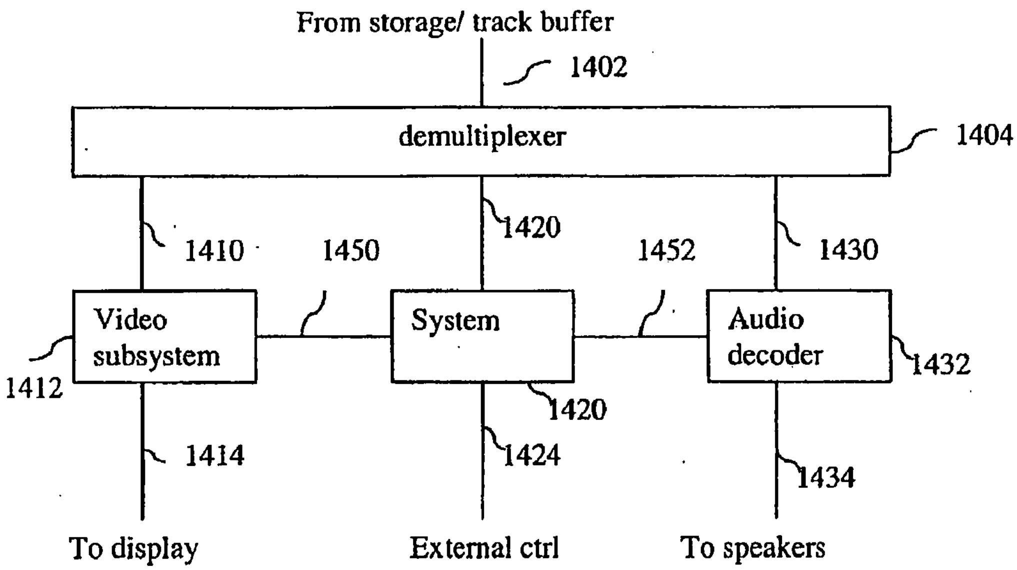



Figure 4b




PRIOR ART

Figure 4c




**PRIOR ART**

**Figure 4d**

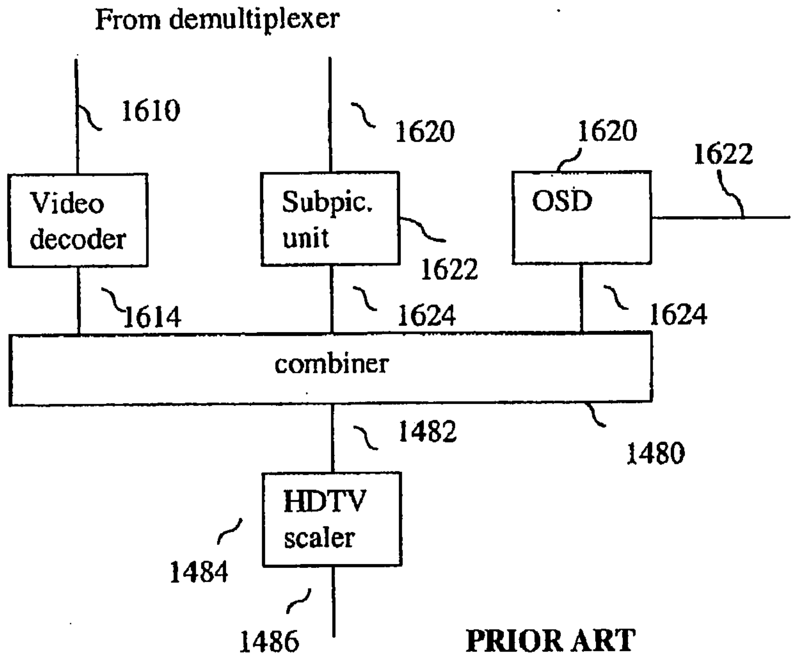



Figure 4e

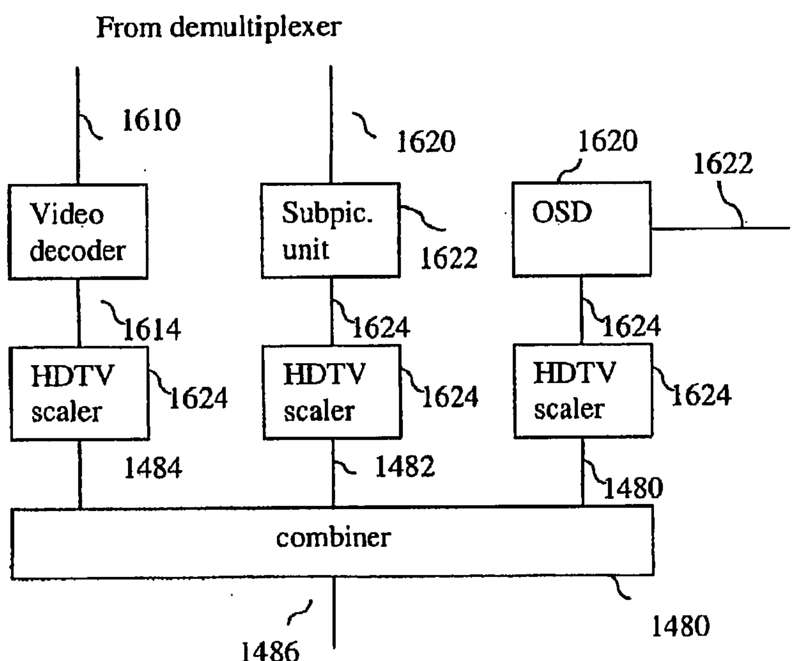



Figure 1c

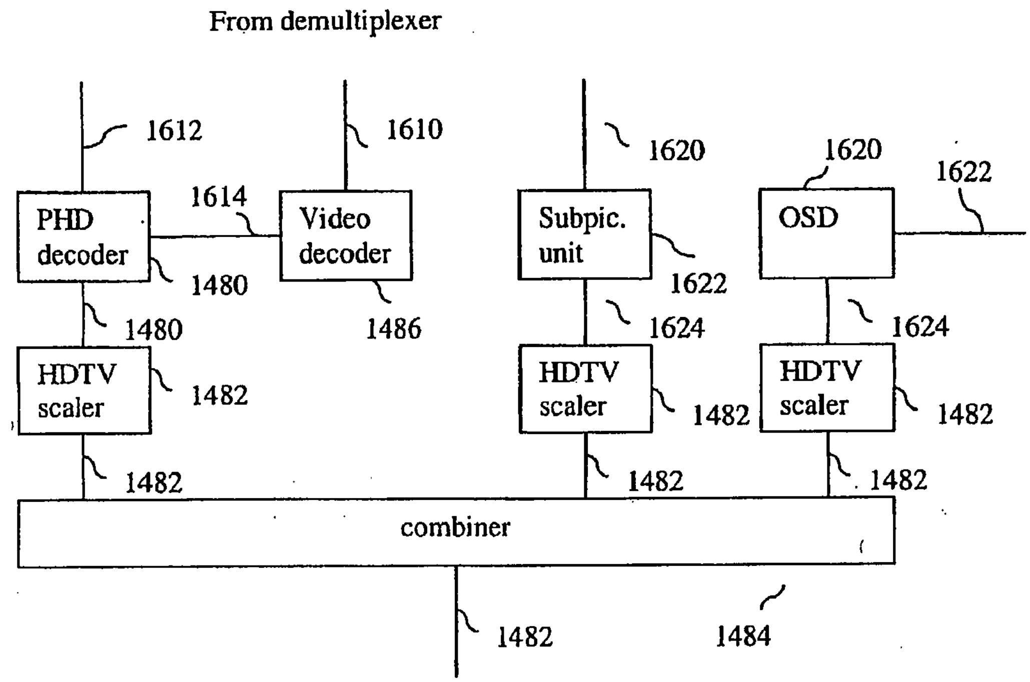
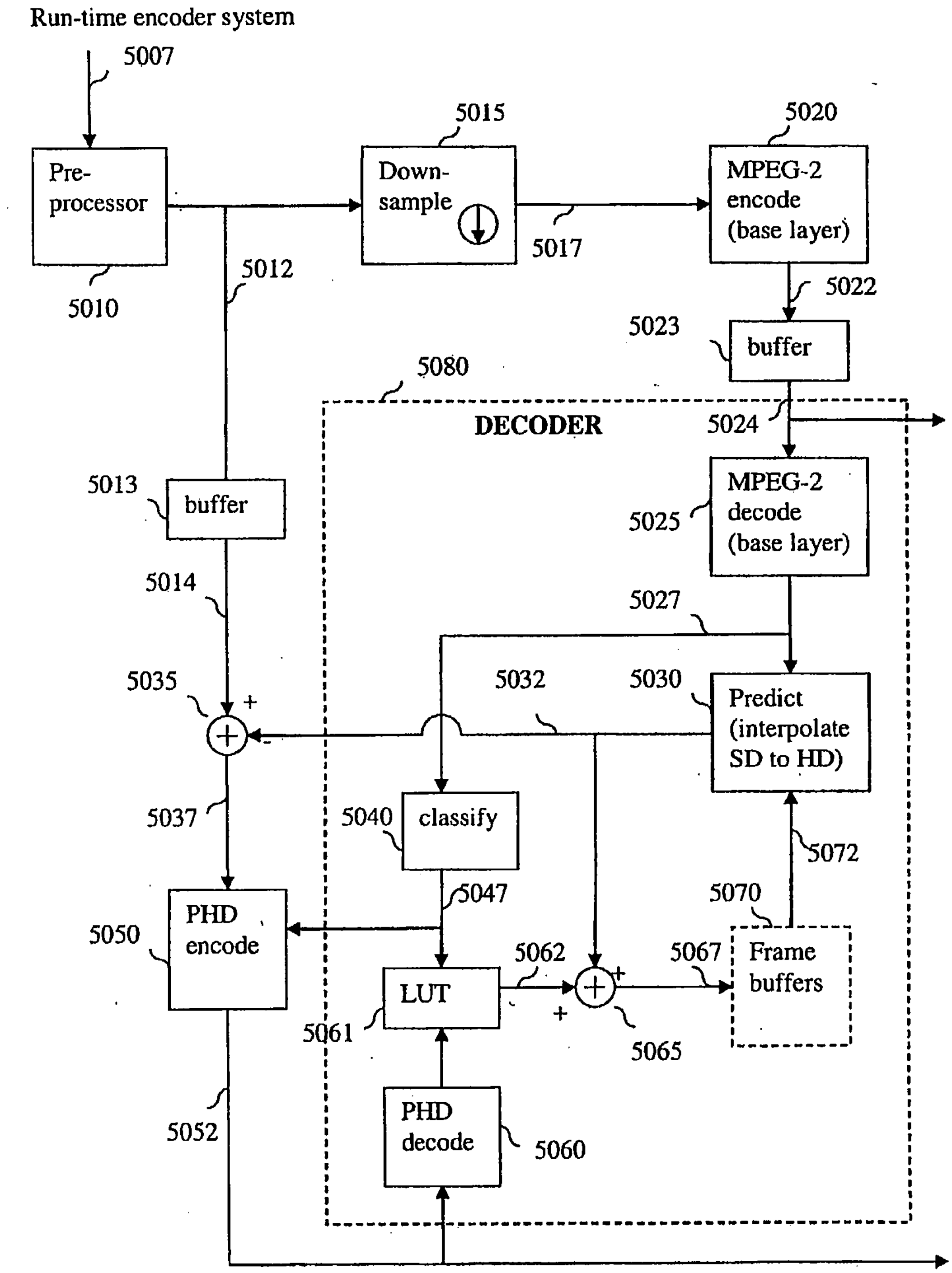

From demultiplexer
1612
1610
1620
1620
1622
1614
PHD decoder
Video decoder
Subpic. unit
OSD
1480
1622
1486
1480
1624
1624
HDTV scaler
HDTV scaler
HDTV scaler
1482
1482
1482
1482
1482
1482
combiner
1482
1484

Figure 4f



Run-time encoder system
5007
Pre-processor
5010
5012
Down-sample
5015
5017
MPEG-2 encode (base layer)
5020
5022
5023
buffer
5080
DECODER
5024
5013
buffer
5014
MPEG-2 decode (base layer)
5025
5027
5035
+
-
5032
5030
Predict (interpolate SD to HD)
5037
5040
classify
5072
5047
5070
PHD encode
5050
LUT
5061
5062
+
+
5065
5067
Frame buffers
5052
PHD decode
5060


Figure 5a

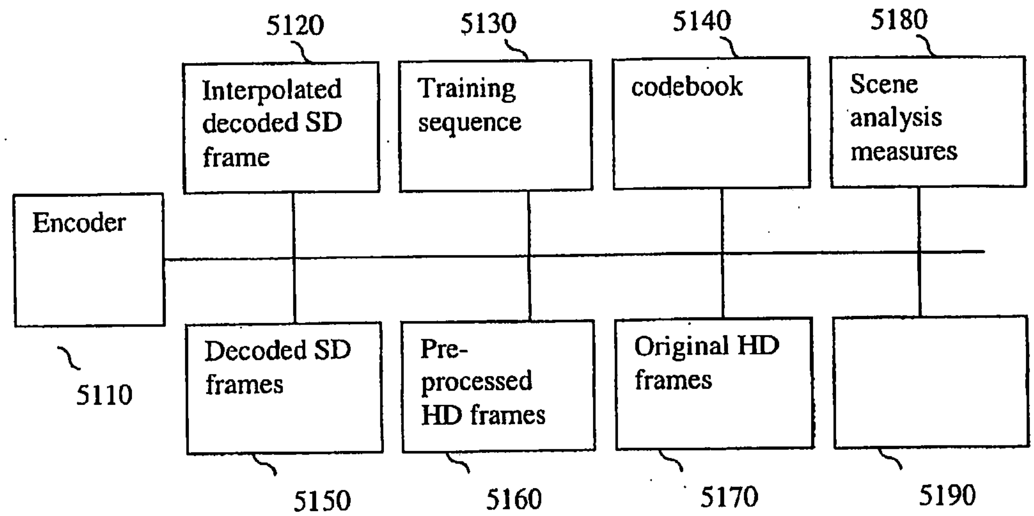

5120
Interpolated decoded SD frame
5130
Training sequence
5140
codebook
5180
Scene analysis measures
Encoder
5110
Decoded SD frames
Pre-processed HD frames
Original HD frames
5150
5160
5170
5190

Figure 5b

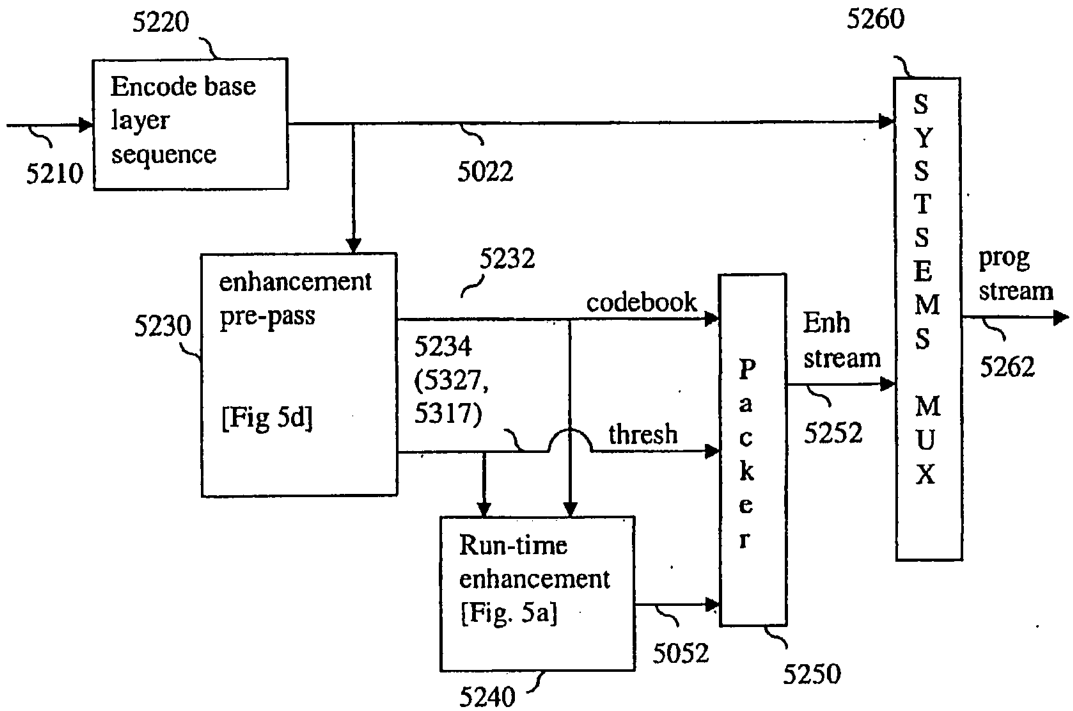



Figure 5c PHD encode steps

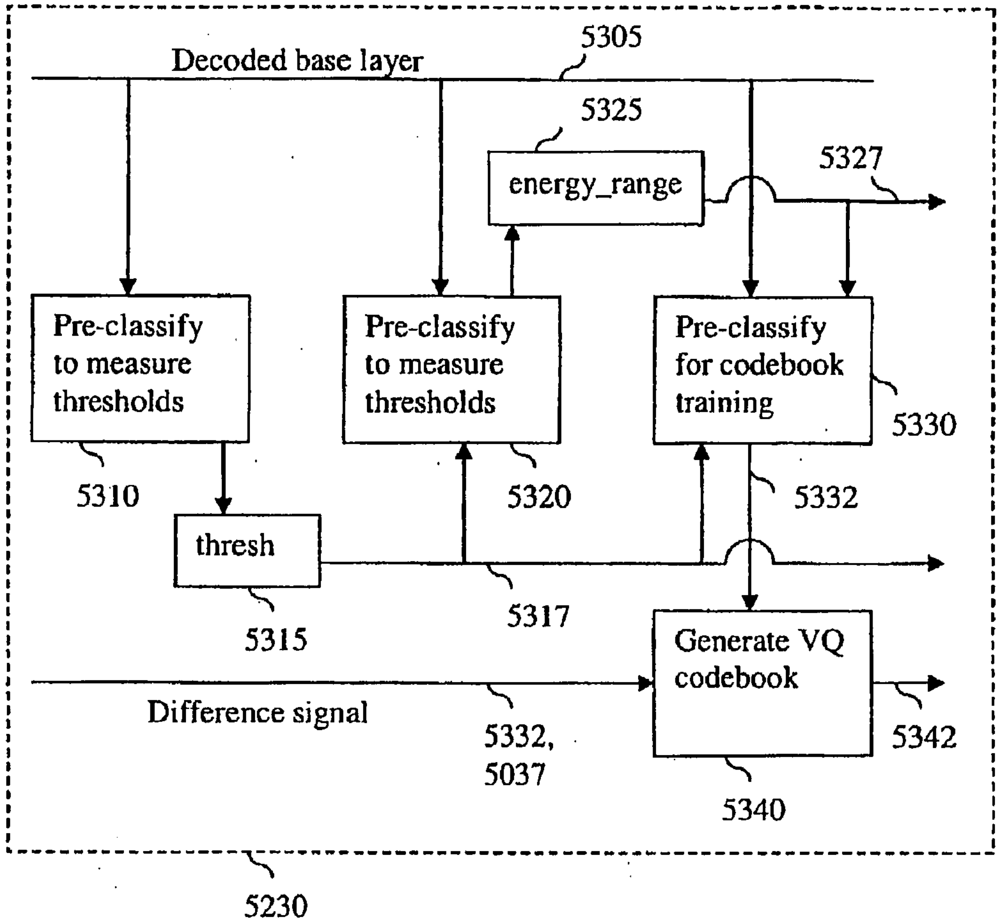



Figure 5d PHD pre-pass operations

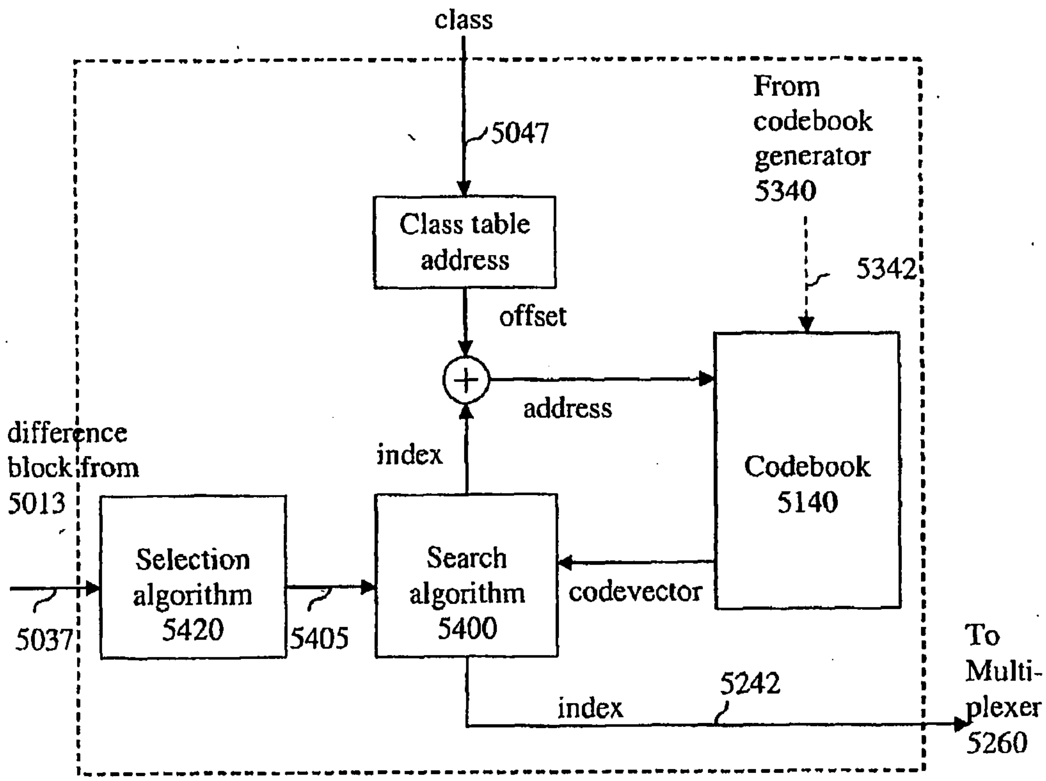



Figure 5e

Authoring figures

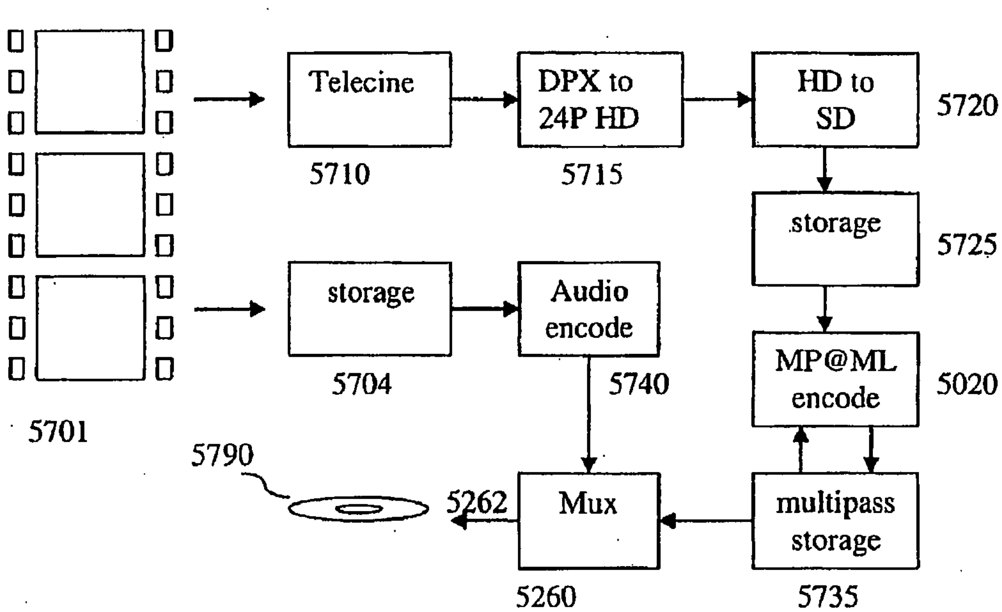



Figure 5h Prior art DVD authoring

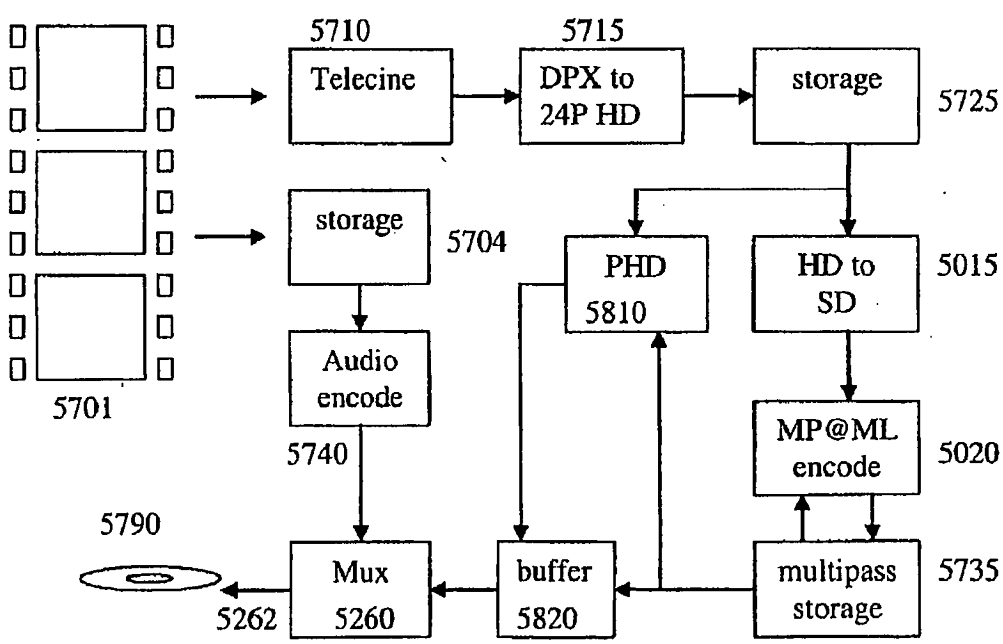



Fig.5i: storage prior to multiplexing disc record.

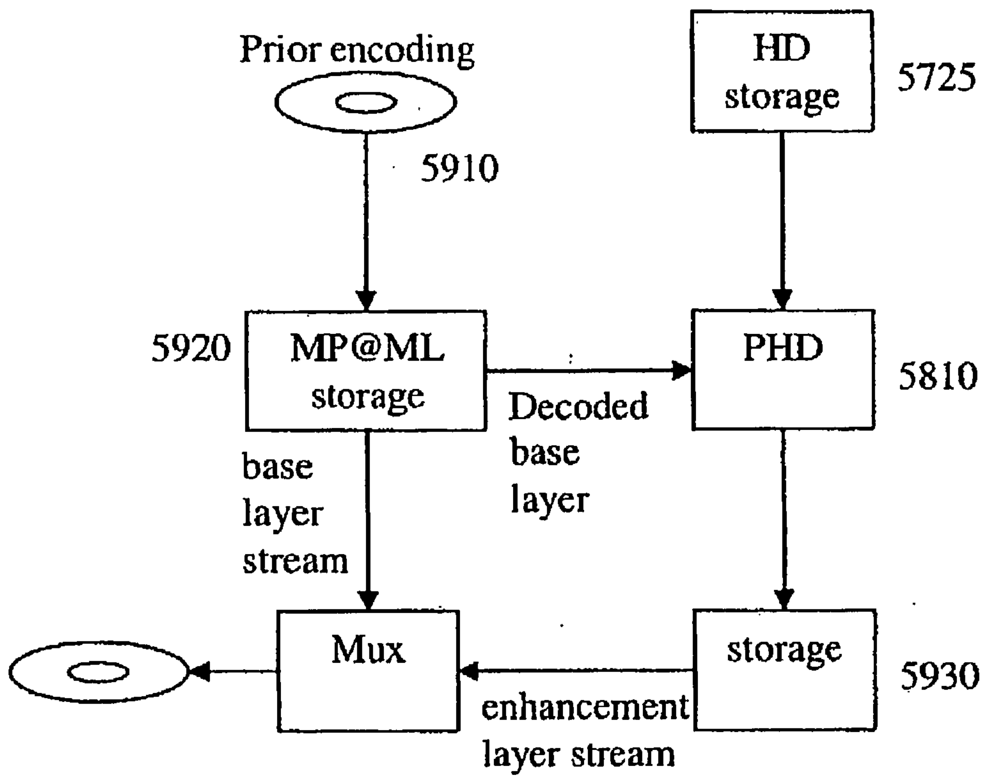



Fig.5j: post authoring.

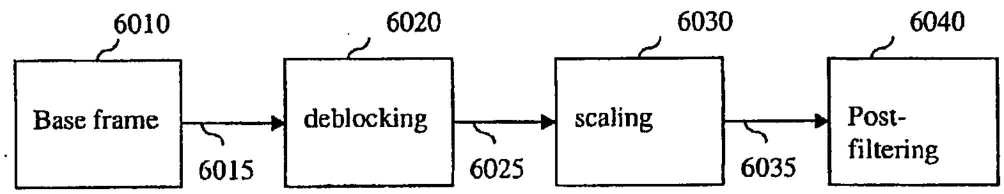



Figure 6a

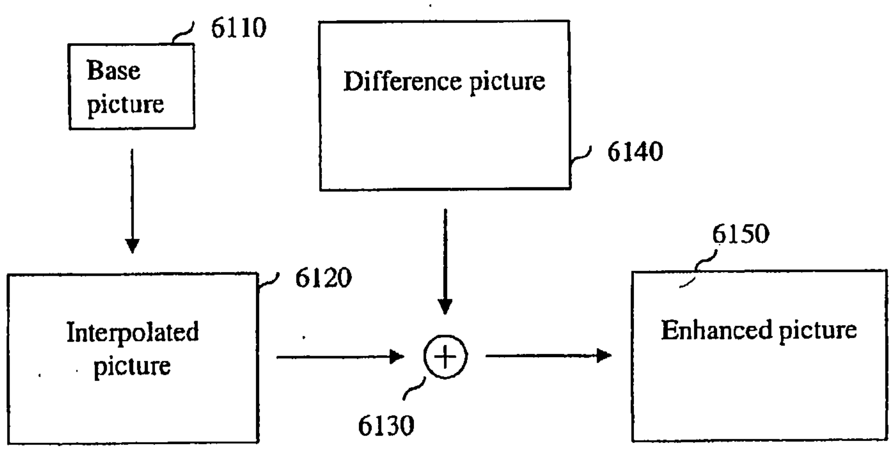
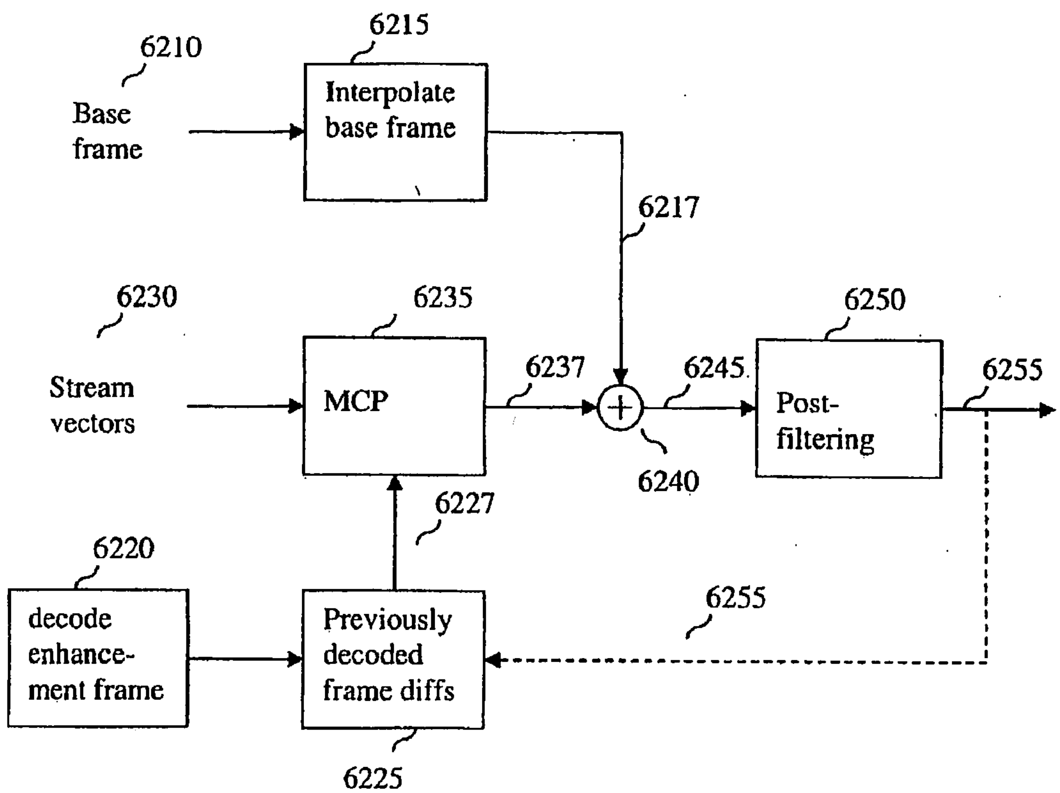



Figure 6b



6210
Base frame
6215
Interpolate base frame
6217
6230
Stream vectors
6235
MCP
6237
6245
6240
6250
Post-filtering
6255
6227
6220
decode enhancement frame
Previously decoded frame diffs
6255
6225


Figure 6c

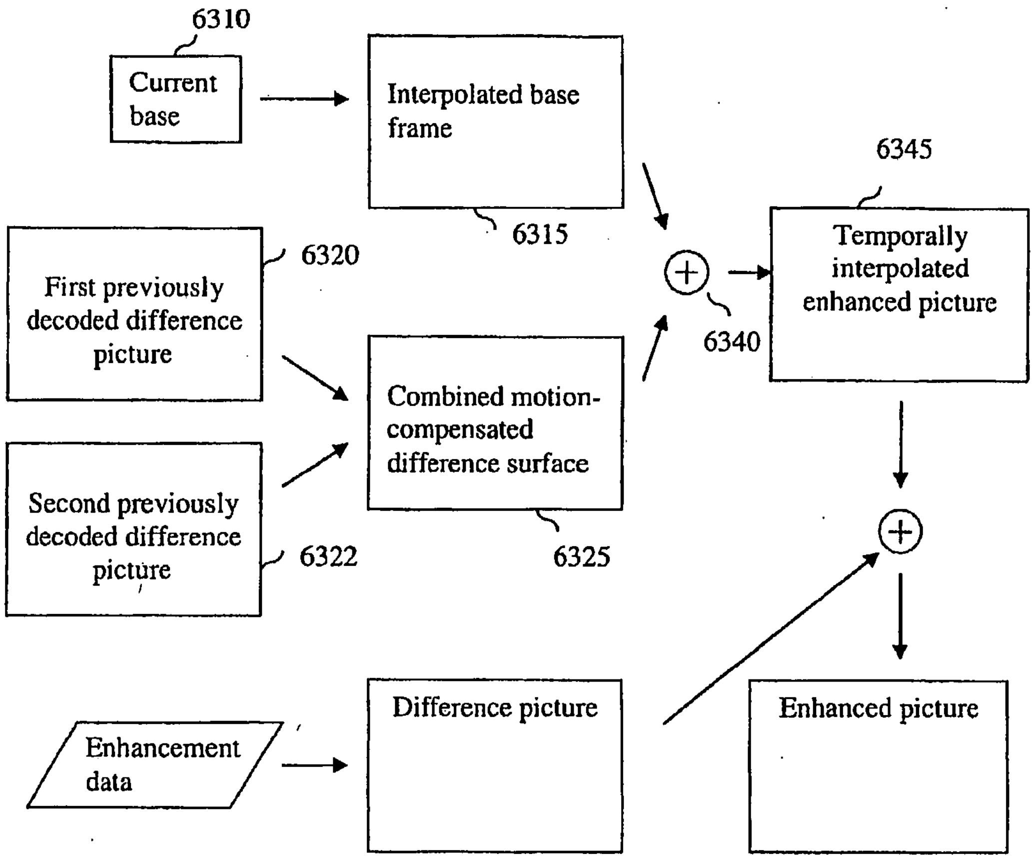

6310
Current base
Interpolated base frame
6315
6345
Temporally interpolated enhanced picture
6340
First previously decoded difference picture
6320
Combined motion-compensated difference surface
Second previously decoded difference picture
6322
6325
Enhancement data
Difference picture
Enhanced picture

Figure 6d

Fig. 7, syntax and semantic definitions of data elements

**Syntax fragments**

```
scene()
{
  scene_code                              32
  scene_number                            24
  n_cbks                                   6
  previous_scene_dependencies              1
  reserved                                 1

  for(i=0;i<n_cbks;i++)
    codebook();

  while( !end_of_scene_code )
  {
    enhancent_picture();
  }

  end_of_scene_code                       32
}

codebook()
{
  codebook_code                           32
  codebook_number                          8
  n_bytes_codebook                        24
  n_classes                                8

  energy_range();                          ?
  thresholds();                            ?

  for(i=0;i<n_classes;i++)
    download_codebook()
}

download_codebook()
{
  cbk_n                                    8
  class_n                                  8
  n_vectors                               16

  for(i=0;i<n_vectors;i++)
    cbk[cbk_n][class_n][i] = vector;
```

```
  stuffing_bits                                   1-7
}

enhancement_picture()
{
  picture_number                                  8
  n_cbk_ud                                        8
  is_picture_enhanced                             1

  for(i=0;i<n_cbk_ud;i++)
    update_codebook();

  if(  is_picture_enhanced )
    for(;;)
      strip()
}

update_codebook()
{
  ud_cbk_n
  ud_class_n
  ud_offset
  n_ud_vec

  for(i=0;i<n_ud_vec;i++)
    cbk[ud_cbk_n][ud_class_n][ud_offset+i] = update_vector;

  stuffing_bits                                   1-7
}

vector()
{
  for(i=0;i<64;i++)
    element[i]                                    8
}

update_vector()
{
  for(i=0;i<64;i++)
    element[i] += diff_element[i]                 VLC
}
```

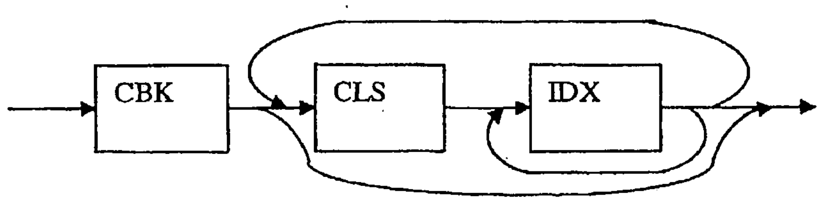



Figure 7a strip diagram

```
strip()
{
  strip_counter                    3
  is_strip_enhanced                1

  y_location                       8
  x_location                       8
  codebook_number                  8
  n_blocks                        16
  class_checksum                  32
  reserved                         8

  if( is_strip_enhanced )
    for(i=0;i<n_blocks;i++)
      enhancement_block[y_location][x_location][i] = index;
}
```

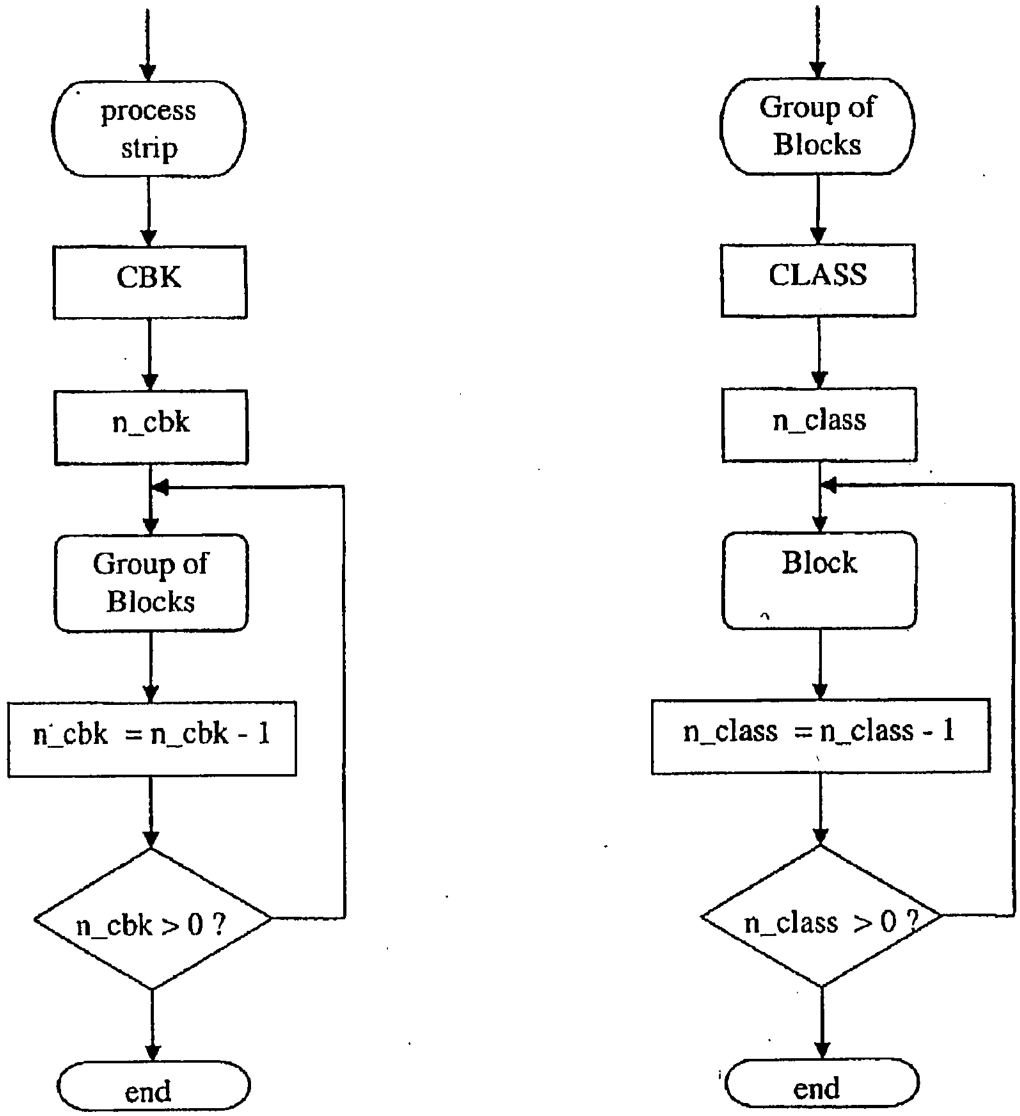



**Figure 7b procedure for parsing a strip()**

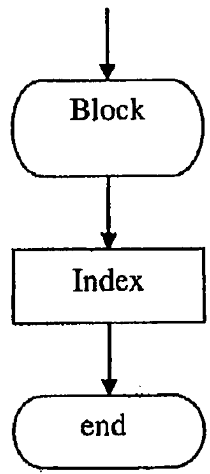



Figure 7c block

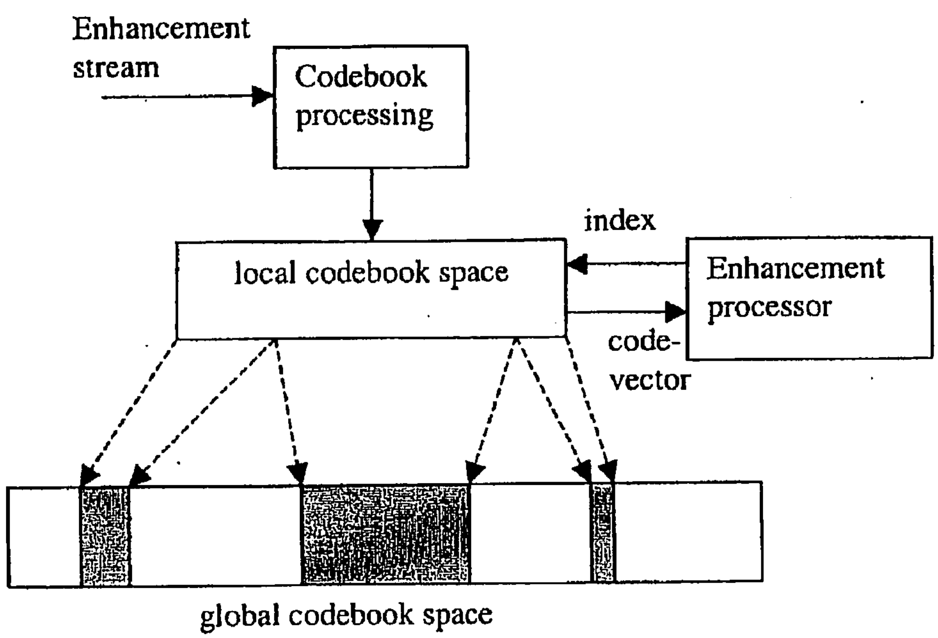



Figure 7d

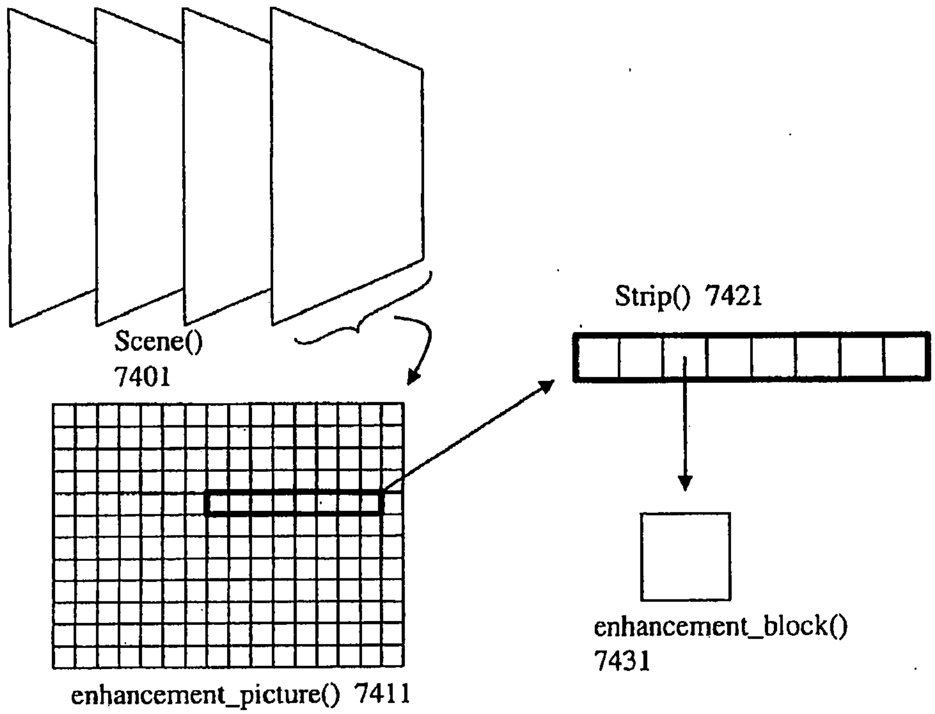



Figure 7e – set of pictures, block delineation within a picture

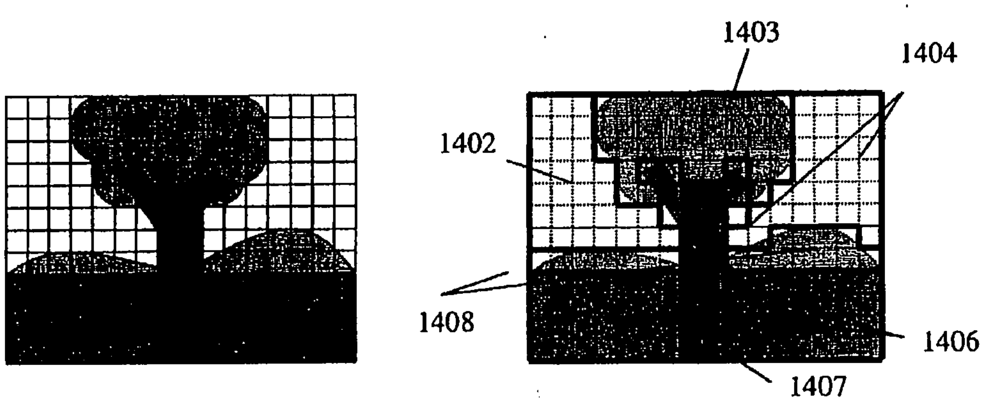



Figure 7f -- codebook selection by content region

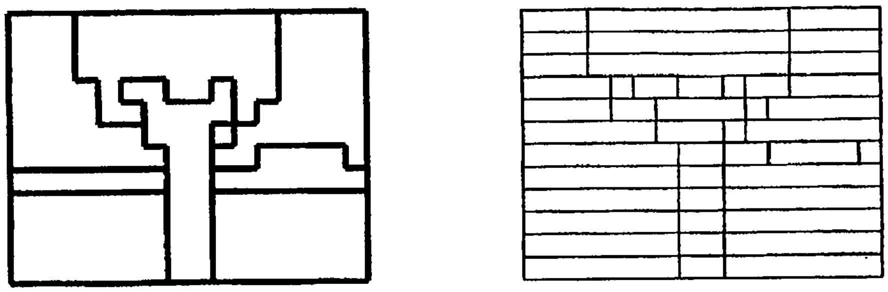


Figure 7g – strip() delineation according to region

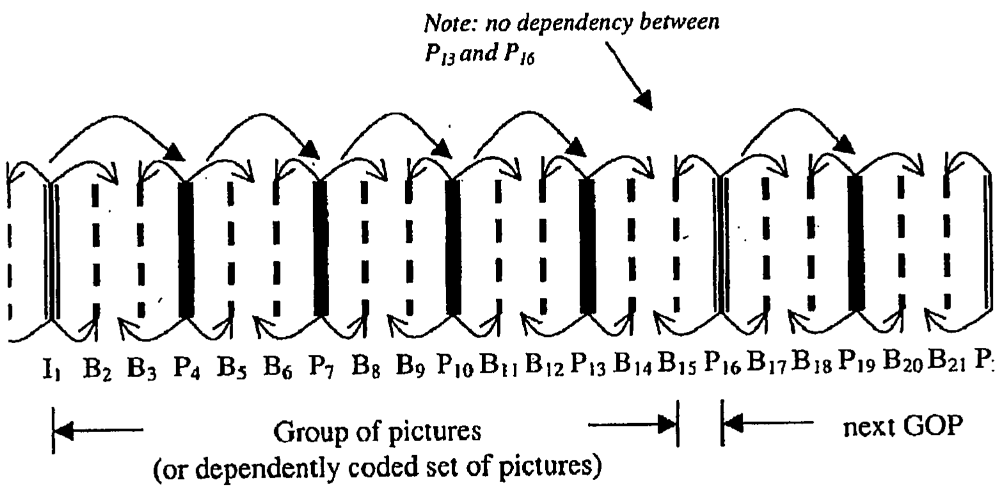


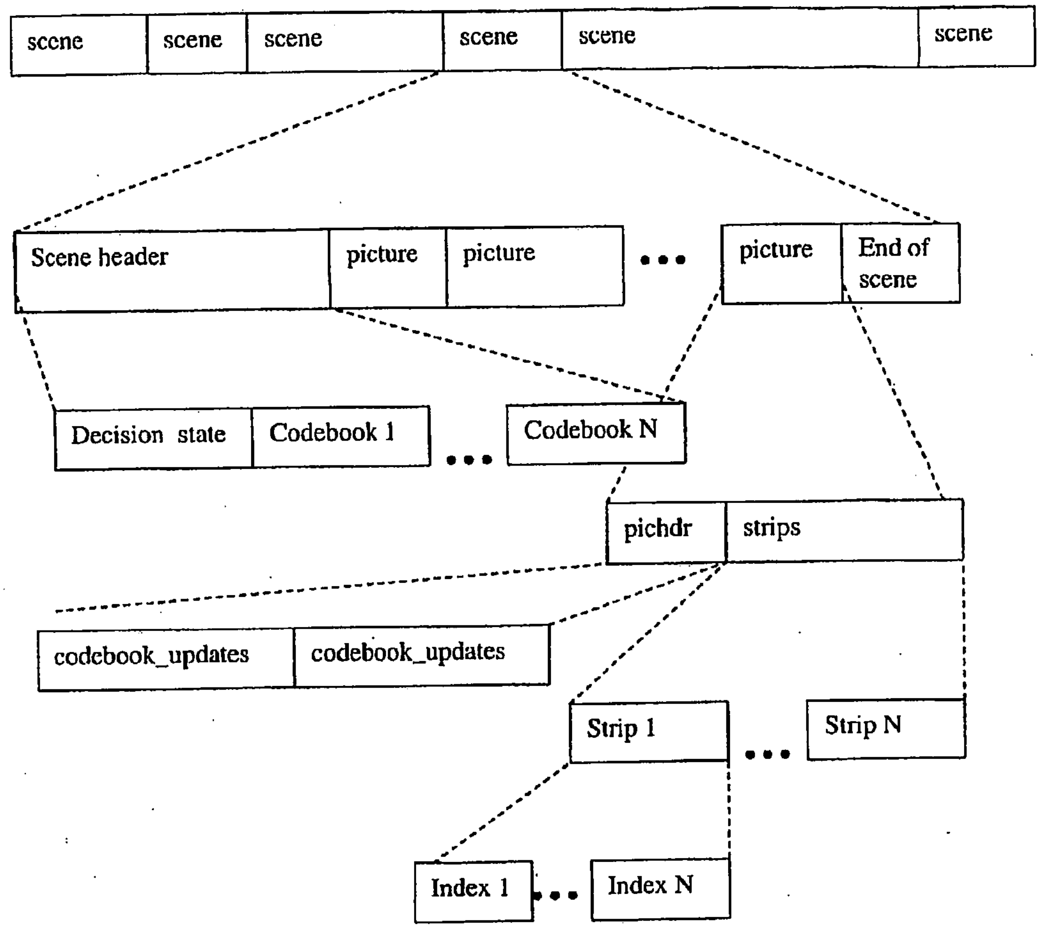

Figure 7h-- Group of dependently coded pictures

Legend:

I : Intra picture

P: Predicted picture

B; Bi-directionally predicted picture


scene
scene
scene
scene
scene
scene
Scene header
picture
picture
picture
End of scene
Decision state
Codebook 1
Codebook N
pichdr
strips
codebook_updates
codebook_updates
Strip 1
Strip N
Index 1
Index N

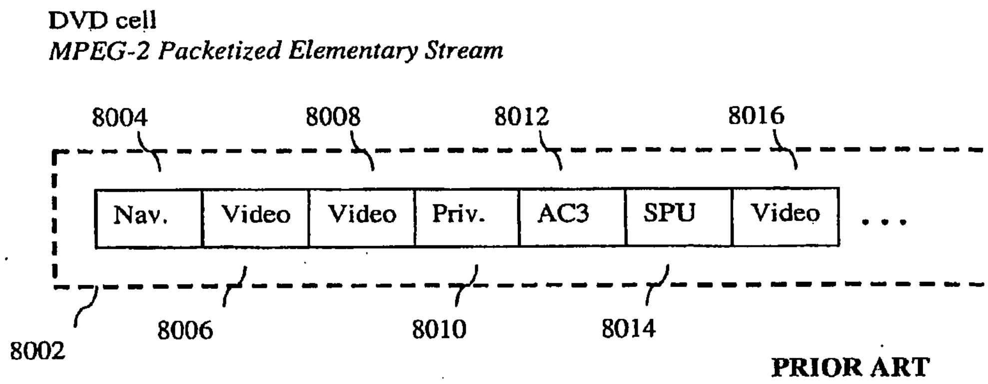



Figure 8a

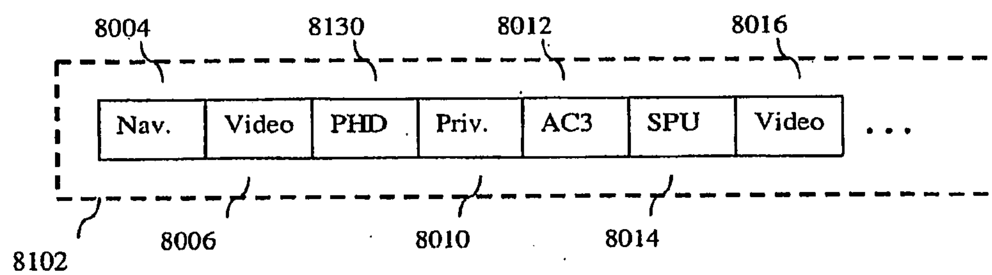



Figure 8b

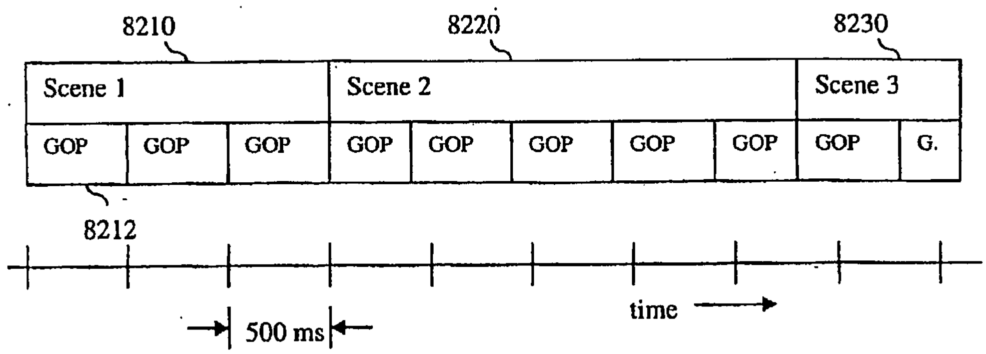



Figure 8c *Prior art: scenes and GOPs*

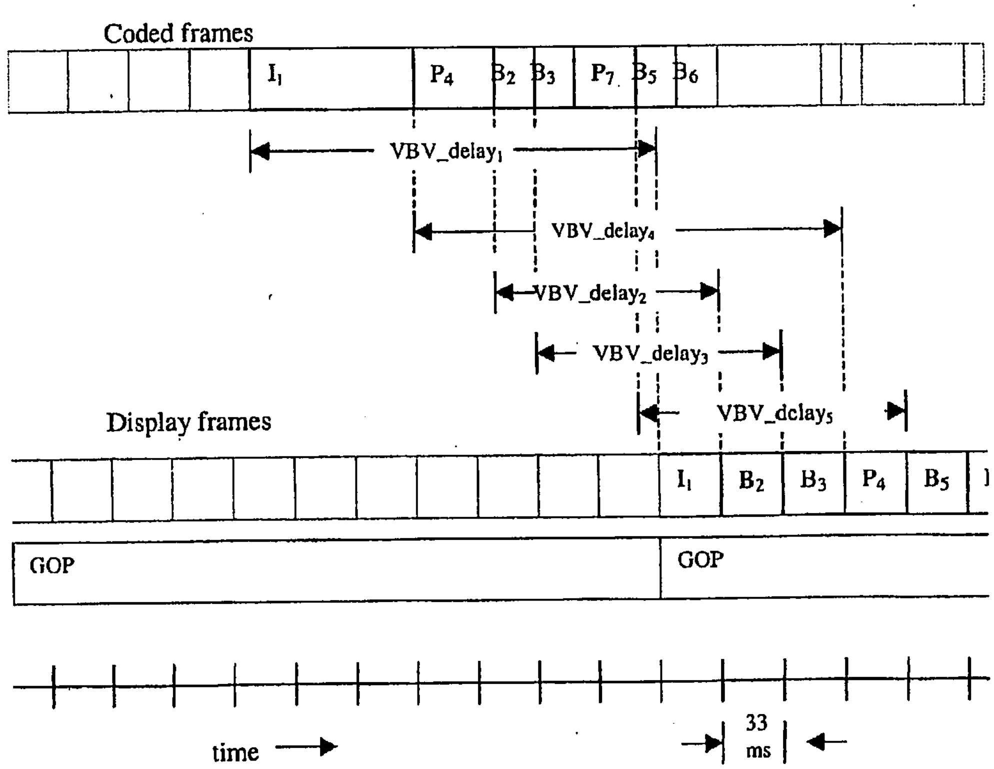



Figure 8d *Prior art relationship between coded frame and display frame times*

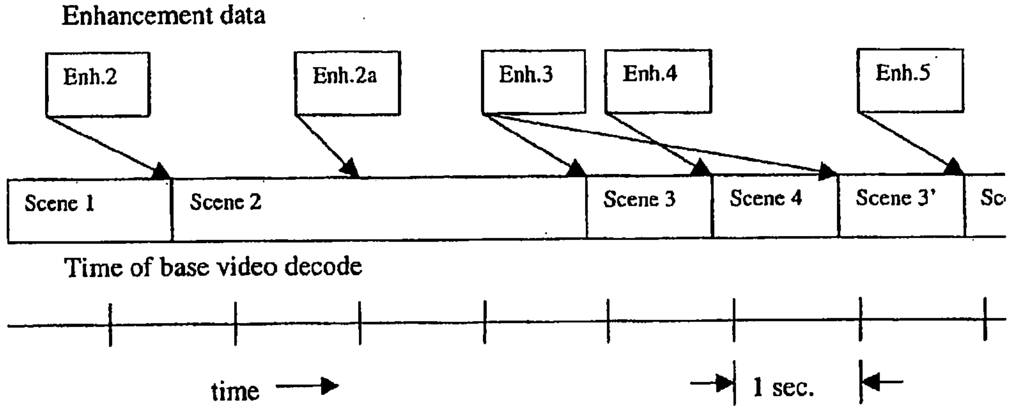



**Figure 8e PHD codebook application periods**

*Legend:*
*Enh2: codebook for scene 2*
*Enh2a: second codebook for scene 2 for random access/resilience purposes due to long scene.*
*Enh3: codebook applied to Scene 3 and Scene 3'. Scene 3' is short enough and close enough in content and time to Scene 3 that only one codebook need by applied.*
*Enh4: codebook for scene 4*
*Enh5: codebook for scene 5 (not shown)*

*Enh1 codebook for Scene 1 is now shown and would be to the left of the diagram.*

## VIDEO INTERPOLATION CODING

### CLAIM FOR PRIORITY

[0001] Applicants claim priority of invention to U.S. Provisional Application 60/384,047, entitled VIDEO INTERPOLATION CODING, filed on May 29, 2002 by the inventors of the present invention.

### RELATED APPLICATIONS

[0002] This application relates to U.S. application Ser. No. ______/______, entitled CLASSIFYING IMAGE AREAS OF A VIDEO SIGNAL, U.S. application Ser. No. ______/______, entitled MAINTAINING A PLURALITY OF CODEBOOKS RELATED TO A VIDEO SIGNAL, and U.S. application Ser. No. ______/______, entitled PREDICTIVE INTERPOLATION OF A VIDEO SIGNAL, each filed concurrently on May 28, 2003 by the inventors of the present invention.

[0003] Pixonics High Definition (PHD) significantly improves perceptual detail of interpolated digital video signals with the aide of a small amount of enhancement side information. In its primary application, PHD renders the appearance of High Definition Television (HDTV) picture quality from a Standard Definition Television (SDTV) coded DVD movie which has been optimized, for example, for a variable bitrate average around 6 mbps (megabits-per-second), while the multiplexed enhancement stream averages approximately 2 mbps.

### BACKGROUND

[0004] In 1953, the NTSC broadcast system added a scalable and backwards-compatible color sub-carrier signal to then widely deployed 525-line black-and-white modulation standard. Newer television receivers that implemented NTSC were equipped to decode the color enhancement signal, and then combine it with the older black-and-white component signal in order to create a full color signal for display. At the same time, neither the installed base of older black-and-white televisions, nor the newer black-and-white only televisions designed with foreknowledge of NTSC would need color decoding circuitry, nor would be noticeably affected by the presence of the color sub-carrier in the modulated signal. Other backwards-compatible schemes followed NTSC.

[0005] Thirty years later, PAL-Plus (ITU-R BT. 1197) added a sub-carrier to the existing PAL format that carries additional vertical definition for letterboxed video signals. Only a few scalable analog video schemes have been deployed, but scalability has been more widely adopted in audio broadcasting. Like FM radio, the North American MTS stereo (BTSC) audio standards for television added a sub-carrier to modulate the stereo difference signal, which when matrix converted back to discrete L+R channels, could be combined in advanced receivers with the mono carrier to provide stereo audio.

[0006] In most cases, greater spectral efficiency would have resulted if the encoding and modulation schemes had been replaced with state-of-the-art methods of the time that provided the same features as the scalable schemes. However, each new incompatible approach would have displaced the installed base of receiving equipment, or required spectrum inefficient simulcasting. Only radical changes in technology, such as the transition from analog to digital broadcast television, have prompted simultaneous broadcasting ("simulcasting") of related content, or outright replacement of older equipment.

[0007] Prior attempts to divide a compressed video signal into concurrent scalable signals containing a base and at least one enhancement layer have been under development since the 1980's. However, unlike analog, no digital scalable scheme has been deployed in commercial practice, largely due to the difficulties and overheads created by the scalable digital signals. The key reason perhaps is found is in the very nature in which the respective analog and digital consumer distribution signals are encoded: analog spectra have regular periods of activity (or inactivity) where the signal can be cleanly partitioned, while digital compressed signals have high entropy and irregular time periods that content is modulated.

[0008] Analog signals contain high degree of redundancy, owing to their intended memory-less receiver design, and can therefore be efficiently sliced into concurrent streams along arbitrary boundaries within the signal structure. Consumer digital video distribution streams such as DVD, ATSC, DVB, Open Cable, etc., however apply the full toolset of MPEG-2 for the coded video representation, removing most of the accessible redundancy within the signal, thereby creating highly variable, long-term coding dependencies within the coded signal. This leaves fewer cleaner dividing points for scalability.

[0009] The sequence structure of different MPEG picture coding types (I, P, B) has a built-in form of temporal scalability, in that the B pictures can be dropped with no consequence to other pictures in the sequence. This is possible due to the rule that no other pictures are dependently coded upon any B picture. However, the instantaneous coded bitrate of pictures varies significantly from one picture to another, so temporal scalable benefits of discrete streams is not provided by a single MPEG bitstream with B-pictures.

[0010] The size of each coded picture is usually related to the content, or rate of change of content in the case of temporally predicted areas of the picture. Scalable streams modulated on discrete carriers, for the purposes of improved broadcast transmission robustness, are traditionally designed for constant payload rates, especially when a single large video signal, such as HDTV, occupies the channel. Variable Bit Rate (VBR) streams provide in practice 20% more efficient bit utilization that especially benefits a statistical multiplex of bitstreams.

[0011] Although digital coded video for consumer distribution is only a recent development, and the distribution mediums are undergoing rapid evolution, such as higher density disks, improved modems, etc., scalable schemes may bridge the transition period between formats.

[0012] The Digital Versatile Disc (DVD), a.k.a. "Digital Video Disc," format is divided into separate physical, file systems, and presentation content specifications. The physical and file formats (Micro-UDF) are common to all applications of DVD (video, audio only, computer file). Video and audio-only have their respective payload specifications that define the different data types that consume the DVD storage volume.

[0013] The video application applies MPEG-2 Packetized Elementary Streams (PES) to multiplex at least three compulsory data types. The compulsory stream types required by DVD Video are: MPEG-2 Main Profile @ Main Level (standard definition only) for the compressed video representation; Dolby AC-3 for compressed audio; a graphic overlay (sub-picture) format; and navigation information to support random access and other trick play modes. Optional audio formats include: raw PCM; DTS; and MPEG-1 Layer II. Because elementary streams are encapsulated in packets, and a systems demultiplexer with buffering is well defined, it is possible for arbitrary streams types to be added in the future, without adversely affecting older players. It is the role of the systems demultiplexer to pass only relevant packets to each data type specific decoder.

[0014] Future supplementary stream types envisioned include "3D" stereo vision, metadata for advanced navigation, additional surround-sound or multilingual audio channels, interactive data, and additional video streams (for supporting alternate camera angles) that employ more efficient, newer generation video compression tools.

[0015] Two major means exist for multiplexing supplementary data, such as enhancement stream information of this invention, in a backwards-compatible manner. These means are not only common to DVD, but many other storage mediums and transmission types including D-VHS, Direct Broadcast Satellite (DBS), digital terrestrial television (ATSC & DVB-T), Open Cable, among others. As the first common means, the systems stream layer multiplex described above is the most robust solution since the systems demultiplexer, which comprises a parser and buffer, is capable of processing streams at highly variable rates without consequence to other stream types multiplexed within the same systems stream. Further, the header of these system packets carry a unique Regiestered ID (RID) that, provided they are properly observed by the common users of the systems language, uniquely identify the stream type so that no other data type could be confused for another, including those types defined in future. SMPTE-RA is such an organization charged with the responsibility of tracking the RID values.

[0016] The other, second means to transport supplementary data, such as enhancement data of the invention, is to embed such data within the elementary video stream. The specific such mechanisms available to MPEG-1 and MPEG-2 include user_data( ), extension start codes, reserved start codes. Other coding languages also have their own means of embedding such information within the video bitstream. These mechanisms have been traditionally employed to carry low-bandwidth data such as closed captioning and teletext. Embedded extensions provides a simple, automatic means of associating the supplementary data with the intended picture the supplementary data relates to since these embedded transport mechanisms exist within the data structure of the corresponding compressed video frame. Thus, if a segment of enhancement data is found within a particular coded picture, then it is straight-forward for a semantic rule to assume that such data relates to the coded picture with which the data was embedded. Also, there is no recognized registration authority for these embedded extensions, and thus collisions between users of such mechanisms can arise, and second that the supplementary data must be kept to a minimum data rate. ATSC and DVD have made attempts to create unique bit patterns that essentially serve as the headers and identifiers of these extensions, and register the ID's, but it is not always possible to take a DVD bitstream and have it translate directly to an ATSC stream.

[0017] Any future data stream or stream type therefore should have a unique stream identifier registered with, for example, SMPTE-RA, ATSC, DVD, DVB, OpenCable, etc. The DVD author may then create a Packetized Elementary Stream with one or more elementary streams of the this type.

[0018] Although the sample dimensions of the standard definition format defined by the DVD video specification are limited to 720×480 and 720×576 (NTSC and PAL formats, respectively), the actual content of samples may be significantly less due to a variety of reasons.

[0019] The foremost reason is the "Kell Factor," which effectively limits the vertical content to approximately somewhere between ⅔ and ¾ response. Interlaced displays have a perceived vertical rendering limit between 300 and 400 vertical lines out of a total possible 480 lines of content. DVD video titles are targeted primarily towards traditional 480i or 576i displays associated with respective NTSC and PAL receivers, rather than more recent 480p or computer monitors that are inherently progressive (the meaning of "p" in 480p). A detailed description of the Kell Factor can be found in the books "Television Engineering Handbook" by Wilkonson et al, and "Color Spaces" by Charles Poynton. A vertical reduction of content is also a certain measure to avoid the interlace flicker problem implied by the Kell Factor. Several stages, such as "film-to-tape" transfer, can reduce content detail. Interlace cameras often employ lenses with an intentional vertical low-pass filter.

[0020] Other, economical reasons favor moderate content reduction. Pre-processing stages, especially low-pass filtering, prior to the MPEG video encoder can reduce the amount of detail that would need to be prescribed by the video bitstream. Assuming, the vertical content is already filtered for anti-flicker (Kell Factor), filtering along the horizontal direction can further lower the average rate of the coded bitstream by a factor approximately proportional to the strength of the filtering. A 135 minute long movie would have an average bitrate of 4 mbps if it were to consume the full payload of a single-sided, single-layer DVD (volume of 4.7 billion bytes). However, encoding of 720×480 interlace signals have been shown to require sustained bitrates as high as 7 or 8 mbps to achieve transparent or just-noticeable-difference (JND) quality, even with a well-designed encoder. Without pre-filtering, a 4 mbps DVD movie would likely otherwise exhibit significant visible compression artifacts. The measured spectral content of many DVD tiles is effectively less than 500 horizontal lines wide (out of 720), and thus the total product (assuming 350 vertical lines) is only approximately half of the potential information that can be expressed in a 720×480 sample lattice. It is not surprising then that such content can fit into half the bitrate implied at least superficially by the sample lattice dimensions.

[0021] The impact of this softening is minimized by the fact that most 480i television monitors are not capable of rendering details within the Nyquist limits of 720×480. The displays are likely optimized for an effective resolution of 500×350 or worse. Potentially, anti-flicker filters, as commonly found in computer-to-television format converters,

could be included in every DVD decoder or player box, thus allowing true 480 "p" content to be encoded on all DVD video discs. Such a useful feature was neither given as a mandate nor suggested as an option in the original DVD video specification. The DVD format was essentially seen as a means to deliver the best standard definition signals of the time to consumers.

[0022] Prior art interpolation methods can interpolate a standard definition video signal to, for example, a high definition display, but do not add or restore content beyond the limitations of the standard-definition sampling lattice. Prior art methods include, from simplest to most complex: sample replication ("zero order hold"), bi-linear interpolation, poly-phase filters, spline fitting, POCS (Projection on Convex Sets), and Bayesian estimation. Inter-frame methods such as super-resolution attempt to fuse sub-pixel (or "subsample") detail that has been scattered over several pictures by aliasing and other diffusion methods, and can in fact restore definition above the Nyquist limit implied by the standard definition sampling lattice. However such schemes are computationally expensive, non-linear, and do not always yield consistent quality gains frame-to-frame.

[0023] The essential advantage of a high-resolution representation is that it is able to convey more of the actual detail of a given content than a low-resolution representation. The motivation of proving more detail to the viewer is that it improves enjoyment of the content, such as the quality difference experienced by viewers between the VHS and DVD formats.

[0024] High Definition Television (HDTV) signal encoding formats are a direct attempt to bring truly improved definition, and detail, inexpensively to consumers. Modem HDTV formats range from 480p up to 1080p. This range implies that content rendered at such resolutions has anywhere from two to six times the definition as the traditional, and usually diluted, standard definition content. The encoded bitrate would also be correspondingly two to six times higher. Such an increased bitrate would not fit onto modem DVD volumes with the modem MPEG-2 video coding language. Modem DVDs already utilize both layers, and have only enough room left over for a few short extras such as documentaries and movie trailers.

[0025] Either the compression method or the storage capacity of the disc would have to improve to match as the increase in definition and corresponding bitrate of HDTV. Fortunately both storage and coding gains have been realized. For example, H.264 (a.k.a. MPEG-4 Part 10 "Advanced Video Coder") has provided a nominal 2× gain in coding efficiency over MPEG-2. Meanwhile, blue-laser recording has increased disc storage capacity by at least 3× over the original red-laser DVD physical format. The minimal combined coding and physical storage gain factor of 6:1 means that it is possible to place an entire HDTV movie on a single-sided, single-layer disc, with room to spare.

[0026] A high-definition format signal can be expressed independently (simulcast) or dependently (scalable) with respect to a standard-definition signal. The simulcast method codes the standard definition and high definition versions of the content as if they were separate, unrelated streams. Streams that are entirely independent of each other may be multiplexed together, or transmitted or stored on separate mediums, carriers, and other means of delivery. The scalable approach requires the base stream (standard definition) to be first decoded, usually one frame at a time, by the receiver, and then the enhancement stream (which generally contains the difference information between the high definition and standard definition signals) to be decoded and combined with the frame. This may be done piecewise, as for example, each area of the base picture may be decoded just in time prior to the addition of the enhancement data. Many implementation schedules between the base and enhancement steps are possible.

[0027] The simulcast approach is cleaner, and can be more efficient than enhancement coding if the tools and bitrate ratios between the two are not tuned properly. Empirical data suggests that some balance of rates should exist between the base and enhancement layers in order to achieve optimized utilization of bits. Thus if a data rate is required to achieve some picture quality for the base layer established by the installed base of DVD players, for example, then the enhancement layer may require significant more bits in order to achieve a substantial improvement in definition.

[0028] In order to lower the bitrate of the enhancement layer, several tricks can be applied that would not noticeably impact quality. For example, the frequency of intra pictures can be decreased, but at the tradeoff of reduced robustness to errors, greater IDCT drift accumulation, and reduced random access frequency.

[0029] Previous scalable coding solutions have not been deployed in main-stream consumer delivery mediums, although some forms of scalability have been successfully applied to internet streaming. With the exception of temporal scalability (**FIG. 2***e*) that is inherently built-in all MPEG bitstreams that utilize B-frames, the spatial scalable scheme (**FIG. 2***d*), SNR scalable (**FIG. 2***c*) and Data Partitioning schemes documented in the MPEG-2 standard have all incurred a coding efficiency penalty rendering scalable coding efficiency little better, or even worse, than the total bandwidth consumed by the simulcast approach (**FIG. 2***b*). The reasons behind the penalties have not been adequately documented, but some of the known factors include: excessive block syntax overhead incurred when describing small enhancements, and re-circulation of quantization noise between the base and enhancement layers.

[0030] **FIG. 2***a* establishes the basic template where, in subsequent figures, the different scalable coding approaches most fundamentally differ in their structure and partitioning. Bitstream Processing (BP) **2010** includes those traditional serially dependent operations that have a varying density of data and hence variable complexity per coding unit, such as stream parsing, Variable Length Decoding (VLD), Run-Length Decoding (RLD), header decoding. Inverse Quantization (IQ) is sometimes placed in the BP category if only the non-zero transform coefficients are processed rather applying a matrix operation upon all coefficients. Digital signal processing (DSP) **2020** operations however tend to be parallelizable (e.g. SIMD scalable), and have regular operations and complexity. DSP includes IDCT (Inverse Discrete Cosine Transform) and MCP (Motion Compensated Prediction). Reconstructed blocks **2025** are stored **2030** for later display processing (4:2:0 to 4:2:2 conversion, image scaling, field and frame repeats) **2040**, and to serve as reference for prediction **2031**. From the bitstream **2005**, the BP **2010** produces Intermediate decoded bitstream **2015** comprising

arrays of transform coefficients, reconstructed motion vectors, and other directives that when combined and processed through DSP produce the reconstructed signal **2025**.

[0031] **FIG. 2***b* demonstrates the "simulcast" case of two independent streams and decoders that optionally, through multiplexer **2136**, feed the second display processor **2140**. The most typical application fitting the **FIG. 2***b* paradigm is a first decoder system for SDTV, and a second decoder system for HDTV. Notably, the second decoder's BP **2110** and DSP **2120** stages do not depend upon state from the first decoder.

[0032] The scalable schemes are best distinguished by what processing stages and intermediate data they relate with the base layer. The relation point is primarily application-driven. **FIG. 2***c* illustrates frequency layering, where the relation point occurs at the symbol stages prior to DSP. (symbols are an alternate name for bitstream elements). In block based transform coding paradigms, the symbol stream is predominately in the frequency domain, hence frequency layering. The enhanced intermediate decoded symbols **2215** combined with the intermediate decoded base symbols **2015** creates a third intermediate symbol stream **2217** that is forward-compatible decodable, in this example, by the base layer DSP decoder **2220**. The combined stream appears as an ordinary base layer stream with increased properties (bitrate, frame rate, etc.) over the base stream **2005**. Alternatively, the enhanced DSP decoder could have tools not present in the base layer decoder DSP, and **2217** depending on the tools combination and performance level, therefore only backward-compatible (assuming the enhanced DSP is a superset of the base DSP). SNR scalability and Data partitioning are two known cases offrequency layering that produce forward-compatible intermediate data streams **2217** decodable by base layer DSP stages **2020**. Frequency layering is generally chosen for robustness over communications mediums.

[0033] In a forward-compatible application example of frequency layering, detailed frequency coefficients that could be added directly to the DCT coefficient block would be encoded in the enhancement stream, and added **2216** to the coefficients **2015** to produce a higher fidelity reconstructed signal **2225**. The combined stream **2217** resembles a plausible base layer bitstream coded at a higher rate, hence the forward compatible designation. Alternatively, a backward-compatible example would be an enhancement stream that inserted extra chrominance blocks into the bitstream in a format only decodable by the enhanced DSP decoder. The original Progressive JPEG mode and the more recent JPEG-2000 are examples of frequency layering.

[0034] Spatial scalability falls into the second major scalable coding category, spatial layering, whose basic decoding architecture as shown in **FIG. 2***d*. The spatial scalability paradigm exploits the base layer spatial-domain reconstruction **2025** as a predictor for the enhanced reconstruction signal **2327**, much like previously reconstructed pictures serve as reference **2031** for future pictures (only the reference pictures are, as an intermediate step, scaled in resolution). A typical application would have the base layer contain a standard definition (SDTV) signal, while the enhancement layer would encode the difference between the scaled high definition (HDTV) and standard definition reconstruction **2025** scaled to match the lattice of **2325**.

[0035] Spatial layering is generally chosen for scaled decoder complexity, but also serves to improve robustness over communications mediums when the smaller base layer bitstream is better protected against errors in the communications channel or storage medium.

[0036] A third scalability category is temporal layering, where the base layer produces a discrete set of frames, and an enhancement layer adds additional frames that can be multiplexed (in between) the base layer frames. An example application is a base layer bitstream consisting of only I and P pictures could be decoded independently of an enhancement stream containing only B-pictures, while the B-pictures would be dependent upon the base layer reconstruction, as the I and P frame reconstructions would serve as forward and backward MCP (Motion Compensated Prediction) references. Another application is stereo vision, where the base layer provides the left eye frames, and the enhancement layer predicts the right eye frames from the left eye frames, with additional correction (enhancement) to code the left-right difference.

[0037] Enhancement methods that do not employ side information or any significant enhancement layer stream are applied by default in the conversion of SDTV to HDTV. Interpolation, through scaling and sharpening, a standard definition (SDTV) signal to a high definition (HDTV) signal is a method to simulate high definition content, necessary to display SDTV on a high definition monitor. Although the result will not look as good as genuine HDTV content, certain scaling or interpolation algorithms do a much better job than others, as some algorithms better model the differences between a HDTV and SDTV representation of the same content. Edges and textures can be carefully sharpened to provide some of the appearance of HDTV, but will at the same time look artificial since the interpolation algorithm will not sufficiently estimate the true HDTV from the content. Plausible detail patterns can be substituted, but may also retain a synthetic look upon close examination.

[0038] Many methods falling under the genre of super-resolution can partially restore HDTV detail from an SDTV signal under special circumstances, although to do so requires careful and complex motion compensated interpolation since the gain is realized by solving for detail that have been mixed over several pictures through iterative mathematical operations. Superresolution tools require sub-pixel motion compensated precision, similar to that found in newer video coders, and with processing at sub-pixel granularity rather than whole blocks. Thus, instead of one motion vector for every 8×8 block (every 64 pixels), there would be one to four motion vectors generated by the superresolution restoration algorithm at the receiver for every high-definition pixel. Optimization techniques can reduce this complexity, but the end complexity would nonetheless exceed the combined decoding and post-processing complexity of the most advanced consumer video systems. In an effort to improve stability of the restored image, and reduce implementation costs, several approaches have been investigated by researchers to restore high resolution from a combination of a lower resolution image and side information or explicit knowledge available only to the encoder.

[0039] Gersho's 1990 publication *"non-linear VQ interpolation . . . "*[Gersho90] first proposes to interpolate lower resolution still images by means of Vector Quantization (VQ) codebooks (**2410** and **2516**) trained on their original higher resolution image counterparts. Prior interpolation

methods, such as multi-tap polyphase filter banks, generate the interpolated image sample-by-sample (or point-wise) where data is fitted to a model of the interpolated signal through convolution with curves derived from the model. The model is typically a sinc function. Gersho's interpolation procedure (**FIG. 2*f***) closely resembles block coding, where picture (example shown in **FIG. 7*e***) is divided into a grid of input blocks similar to the grid **7411**. Each block (whose relationship to the grid **7411** is demonstrated by block **7431**) in signal **2506** may be processed independently of other blocks within the same picture. The mapping stage **2504** models some form of distortion such as sub-sampling of the original signal **2502** to the input signal **2506**. It is the goal of the Gersho90 interpolator that the reconstructed block **2518** best approximates the original block **2502** given the information available in the receiver, namely, input block **2506** and previously derived codebooks **2510** and **2516**. Input block **2506** is matched to a best-fit entry within a first codebook **2510**. **FIG. 2*g*** adapts the mapping stage **2604** as a combination of decimation followed by the MPEG encode-decode process, the focus of this disclosure's application. Specifically, the mapping stage is the conversion of an HDTV signal to an SDTV signal (via sub-sampling or decimation) that is then MPEG encoded. While the classic VQ picture coder transmits codebook indices to the receiver, in the nonlinear VQ interpolation application (**FIG. 2*f*** through **2*i***), the first index **2512** of the matching codebook entry in **2510** serves as the index of a corresponding entry in a second codebook **2516**. "Super-resolution" is achieved in that the second codebook contains detail exceeding the detail of the input blocks **2506**. Gersho90 is targeted for the application of image restoration, operating in a receiver that is given the distorted image and codebooks **2510**, **2516**, **2610**, and **2616** trained on content **2502** available only at the transmitter.

**[0040]** Gersho's non-linear VQ interpolation method is applied for image restoration, and therefore places the codebook search matching and index calculation routine at the receiver. In contrast, the typical applications of VQ are for compression systems whose search routine is at the transmitter where indices and the codebooks are generated and transmitted to the receiver. The receiver then uses the transmitted elements to reconstruct the encoded images. While in the Gersho90 design, the index generator **2008** is the receiver, the codebook generator still resides at the transmitter, where the higher resolution source content **2002** upon which C* (**2016**, **2116**) is trained, is available.

**[0041]** The principal step of *Non-linear Interpolative Vector Quantization for Image Restoration* described by [Sheppard97], over the [Gersho90] paper that it builds upon, is the substitution of the first VQ stage (**2508**,**2608**) with a block waveform coder comprising a Discrete Cosine Transform **2904** and transform coefficient Quantization stage **2908**. The quantized coefficients are packed **2912** to form the index **2914** applied to the second codebook **2716**, **2812**. Thus, a frequency domain codebook is created rather than the traditional, spatial domain VQ codebook. The significance of this step is many-fold. First, the codebook search routine is reduced to negligible complexity thanks to the combination of DCT, quantization, and packing stages (**2904**, **2908**, **2912** respectively) that collectively calculate the second codebook index **2712** directly from a combination of quantized DCT coefficients **2906** within the same block **2902**. Prior methods, such as Gersho90, generated the index through a comprehensive spatial domain match tests (similar to the process in **5400**) of many codebook entries (similar to **5140**) to find the best match, where the index **2712** of the best match serves as the index sought by the search routine.

**[0042]** Sheppard further overlaps each input block by a pre-determined number of samples. Thus, a window of samples is formed around the projected area to be interpolated, and the input window steps through the picture at a number of samples smaller than the dimensions of the input block. Alternatively, in a non-overlapping arrangement, the projected and input block dimensions and step increments would be identical. An overlapping arrangement induces a smoothing constraint, resulting in a more accurate mapping of input samples to their output interpolated counterparts. This leads to fewer discontinuities and other artifacts in the resulting interpolated image. However, the greater the overlap, the more processing work must be done in order to scale an image of a given size. For example, in a combination of a 4×4 process block overlapping a 2×2 input block, sixteen samples are processed for every four samples that are interpolated. This is a 4:1 ratio of process bandwidth to input work. In a non-overlapping arrangement, sixteen samples (in a 4×4 block) are produced for every sixteen input samples. The overlapping example given here requires four times as much work per average output sample as the non-overlapping case.

**[0043]** Although the DCT method by Sheppard et al does permit larger codebooks than the NLIVQ methods of Gersho et al, it does not address the cost and design of sending such codebooks to a receiver over a communications or storage medium. The application is a "closed circuit" system, with virtually unlimited resources, for restoring images of similar resolution. Thus, an improved system that is designed specifically targeted for entropy-constrained, real-time transmission and can scale across image resolutions is needed.

**[0044]** DVD

**[0045]** DVD is the first inexpensive medium to deliver to main stream consumers nearly the full quality potential of SDTV. Although a rigid definition of SDTV quality does not exist, the modern definition has settled on "D-1" video—the first recording format to adopt CCIR 601 parameters. SDTV quality has evolved significantly since the first widespread introduction of television in the 1940's, spawning many shades of quality that co-exist today.

**[0046]** In the late 1970's, the first popular consumer distribution format, VHS and Betamax tape, established the most common denominator for standard definition with approximately 250 horizontal luminance lines and a signal-to-noise ratio (SNR) in the lower to mid 40's dB range. Early television broadcasts had similar definition. In the 1980's, television monitors, analog laserdiscs, Super-VHS and the S-Video connector offered consumers improved SD video signals with up to 425 horizontal lines and SNR as high as 50 dB, exceeding the 330 horizontal-line-per-picture-height limit of the broadcast NTSC signal format today.

**[0047]** Starting in 1982, professional video engineering organizations collaborated on the creation of the CCIR 601 discrete signal representation standard for the exchange of digital signals between studio equipment. Although it is only one set of parameters among many possible choices, CCIR 601 effectively established the upper limit for standard

definition at 540 horizontal lines per picture height (on a 4:3 aspect ratio monitor). Applications such as DVD later diluted the same pixel grid to cover a one third wider screen area. Thus the horizontal density on 16:9 anamorphic DVD titles is one third less than standard 4:3 "pan & scan" titles. The CCIR 601 rectangular grid sample lattice was defined as 720 samples per line, with approximately 480 lines per frame at the 30 Hz frame rate most associated with NTSC, and 576 lines at the 25 Hz frame rate of PAL and SECAM. Horizontal line density is calculated as (aspect ratio)*(total lines per picture width). For a 4:3 aspect ratio, the yield is therefore ((4/3)*(720))=540 lines.

[0048] Although technically a signal format, CCIR 601 cultivated its own connotation as the ultimate watermark of "studio quality." By the late 1990's, CCIR 601 parameters were ushered to consumers by the ubiquitous MPEG-2 video standard operating mode, specifically designated "Main Profile @ Main Level or "MP@ML". MPEG-2 MP@ML was adopted as the exclusive operating point by products such as DVD, DBS satellite, and digital cable TV. While the sample dimensions of DVD may be fixed to 720×480 ("NTSC") and 720×576 ("PAL"), the familiar variables such as bitrate (bandwidth), content, and encoder quality very much remain dynamic, and up to the discretion of the content author.

[0049] Concurrent to the end of the SDTV evolution, HDTV started from almost its beginning as a handful of digital formats. SMPTE 274M has become HDTV's ubiquitous analogy for to SDTV's CCIR 601. With 1920 samples-per-line by 1080 lines per frame, and a 16:9 aspect aspect ratio—one third wider than the 4:3 ratio of SDTV—SMPTE 274M meets the canonical requirement that HD be capable of rendering twice the horizontal and vertical detail of SDTV. The second HDTV format, SMPTE 296M, has image dimensions of 1280×720 samples.

[0050] Until all programming is delivered in an HDTV format, there will be a need to convert SDTV signals to fit on HDTV displays. SDTV legacy content may also circulate indefinitely. In order to be displayed on a traditional HDTV display, SDTV signals from sources such as broadcast, VHS, laserdisc, and DVD need to first be up-converted to HDTV. Classic picture scaling interpolation methods, such as many-tap FIR poly-phase filters, have been regarded as the state of the art in practical interpolation methods. However, the interpolated SD signal will still be limited to the detail prescribed in the original SD signal, regardless of the sample density or number of lines of the HD display. Interpolated SD images will often appear blurry compared to their true HD counterparts, and if the interpolated SD images are sharpened, they may simulate some aspect of HD at the risk looking too synthetic.

[0051] One reason for SD content looking better on HD displays comes from the fact that most display devices are incapable of rendering the full detail potential of the signal format they operate upon as input. The HD display has the advantage that details within the SD image that were too fine or subtle to be sufficiently resolved by a SD display can become much more visible when scaled up on the HD display. Early on, however, the interpolation processing and HD display will reach a point of diminishing returns with the quality and detail that can be rendered from an SD signal. In the end, information must be added to the SD signal in order to render true detail beyond the native limits of the SD format. Several enhancement schemes, such as the Spatial Scalable coders of MPEG-2, have been attempted to meet this goal, but none have been deployed in commercial practice due to serious shortcomings.

[0052] Enhancement methods are sensitive to the quality of the base layer signal that they build upon. To optimize the end quality, a balance in bitrate and quality must be struck between the base layer and enhancement layer reconstructions. The enhancement layer should not always spend bits correcting deficiencies of the base layer, while at the same time the base layer should not stray too close to its own point of diminishing returns.

## SUMMARY

[0053] **FIG. 1*a*** shows the conceptual performance of the invention when used as an enhancement coder in conjunction with an MPEG-2 base layer. The perceived quality level $Q_2$ achieved with the PHD/MPEG-2 combination at rate $R_2$ is greater than the quality that would be reached using only MPEG-2 at the same rate $R_2$. In this figure, MPEG expresses quality up to a natural stopping point, where PHD picks up and carries it further at a faster rate (denoted with a higher Q/R slope). The figure expresses that there is a natural dividing point between MPEG-2 and PHD that leads to an overall optimal quality.

[0054] While DVD video may be the first popular consumer format to reach the limits of standard definition, artifacts may still be occasionally visible, even on the best coded discs. Those skilled in the art of video coding are familiar with empirical measures that an MPEG-2 video bitstream can sustain up to 10 million bits per second at transparent quality levels when approximating a CCIR 601 rate standard definition video signal containing complex scenes. Sophisticated pre-processing steps can be carefully applied to reduce the content of the signal in areas or time periods that will not be very well perceived, and therefore reduce coded bitrate for those areas, and/or remove data patterns that would not map to a concise description with the MPEG-2 video coding language. Removal of noise, temporal jitter, and film grain can also help reduce bitrate. Human-assisted coding of difficult scenes is used to make decisions on areas or periods that fail encoder analysis. However, even with these and other optimization steps, the average bitrate will, for film content coded at the quality limits of SDTV, be on the order of 6 to 7 mbps. The reference DVD system, defined by the DVD Forum members and documented in the DVD specification, requires that the DVD player transport and multiplexing mechanism shall indefinitely sustain video rates as high as 9.5 mbps.

[0055] Therefore to bridge the transition between the modern DVD standard definition format, and any new high definition format that employs a combination of new coding methods and new storage mediums (which are not backwards compatible with older means), an improved method of enhancement coding is needed.

[0056] The interpolation error signal is the difference between the interpolated signal and the original signal that the interpolation is attempting to estimate or predict. The interpolation error typically has high concentration of energy along edges of objects, since the edges are most difficult to model accurately with prediction. PHD includes

tools for the efficient coding of the most perceptible detail within the interpolation error signal that represents information lost, for example, during the filtering conversion from the original HD signal to the base layer signal.

[0057] PHD efficiently exploits the base layer video information already available to the receiver, thereby minimizing the amount of enhancement information to be sent. Two principal tools are employed to this end: the classifier, and the predictive interpolator. In a specific instance of the preferred embodiment, classification is applied to the base layer to select sub-tables of a codebook that contains a collection of additive detail block patterns activated by the coded enhancement stream. The overall algorithm is conceptualized in **FIG. 1*b*** through the illustration of data at various stages of transformation as data passes through the PHD decoder.

[0058] The preferred instance of the toolset resembles a block-based video coding language. Difference blocks are first sent within the enhancement bitstream to improve or correct the accuracy of the predicted image. Then, individual blocks are applied to interpolated areas. Small block sizes, such as the preferred embodiment's 4×4 base layer classification block size, offer a reasonable tradeoff between bitrate, implementation complexity, and approximation of picture features and contours. Each 4×4 area in the base layer image has a corresponding 8×8 area in the interpolated image.

[0059] The PHD decoder analyzes the base layer data, through for example the preferred classification methods, and adds enhancement data to the interpolated signal. Many stages of the enhancement process are also guided by analysis conducted on the base layer reconstruction. For example, flat background areas that are determined unworthy of enhancement by the base layer analyzer do not incur the overhead of signaling in the enhancement stream of how those areas should be treated.

[0060] To demonstrate the power of the classification tool, **FIG. 1*c*** shows a small codebook **1210** of image patterns before and after partitioning by classification. Codevectors are sorted by their base patterns in the left column **1210**, and then are grouped into the right boxes (**1220**, **1222**, **1224**, **1226**) according to the base pattern common to each cluster of codevectors. The simplified example has four codevectors per each of the four classes. After clustering, the address space **1212** is effectively cut in half, resulting in a 2-bit index **1221**—half the size of the original 4-bit index **1212** (shown along the left column) needed to uniquely address each codevector. The first two prefix bits of the original 4-bit index are effectively derived from the base layer analyzer.

[0061] To demonstrate the application of the classifier, **FIG. 1*d*** shows the set of classes for a simple picture with one foreground object (tree) and several background areas (sky, mountains, and grass). Each block is assigned a class number in **FIG. 1*d***, and a separate sub-table codevector index in **FIG. 1*e***. The object outlines in **FIG. 1*e*** illustrate the high pass signal of the solid objects in **FIG. 1*d***. The high pass, or "difference" signal, is effectively coded with the blocks in the codebook table.

[0062] Any distinct pattern or set of attributes that can be derived from the base layer, through a combination of operations and analytical stages, and has commonality among a sufficient number of codevectors, can serve as a class. The larger the number of codevectors that share common attributes (such as the example base patterns in **FIG. 1*c***), the greater the reduction of the global address space of the codebook and hence smaller the codevector indices that need to be transmitted to the PHD decoder. In other words, the amount of information that nominally need be sent can first be reduced by partially deriving whatever information possible in the receiver.

[0063] Classification also forces unimportant codevectors that do not strongly fall into any class to merge with like codevectors.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0064] **FIG. 1*a*** is a block diagram showing the performance of the invention.

[0065] **FIG. 1*b*** is a block diagram showing the transformation of data as it passes through a decoder according to the present invention.

[0066] **FIG. 1*c*** shows a codebook of image patterns before and after partitioning by classification.

[0067] **FIG. 1*d*** shows a set of classes for one picture according to the present invention.

[0068] **FIG. 1*e*** shows a sub-table codevector index according to the present invention.

[0069] **FIG. 2*a*** shows a block diagram of single non-scalable stream according to the present invention.

[0070] **FIG. 2*b*** shows a block diagram of two independent streams according to the present invention.

[0071] **FIG. 2*c*** is a block diagram showing frequency layer according to the present invention.

[0072] **FIG. 2*d*** is a block diagram showing special scalability according to the present invention.

[0073] **FIG. 2*e*** a block diagram showing temporal scalability according to the present invention.

[0074] **FIG. 2*f*** is a block diagram showing a Gersho interpolation procedure.

[0075] **FIG. 2*g*** is a block diagram showing a mapping stage having a combination of decimation followed by an MPEG encode/decode process according to the present invention.

[0076] **FIG. 2*h*** is a block diagram showing non-linear interpolation vector quantization according to the present invention.

[0077] **FIG. 2*i*** is a block diagram showing non-linear interpolation vector quantization of MPEG encoded video.

[0078] **FIG. 2*j*** is a block diagram showing index generation steps.

[0079] **FIG. 3*b*** is a block diagram showing the fundamental stages of a classifier according to the present invention.

[0080] **FIG. 3*d*** is a block diagram showing the fundamental stages of a classifier according to an alternate embodiment of the present invention.

**[0081]** **FIG. 3*e*** shows a set of coefficients according to the present invention.

**[0082]** **FIG. 3*f*** is a flow chart showing the classification process according to the present invention.

**[0083]** **FIG. 3*g*** is a flow chart showing the state realization of a decision tree.

**[0084]** **FIG. 3*h*** is a block diagram of a state machine according to the present invention.

**[0085]** **FIG. 4*a*** is a block diagram showing a conventional spatial scalable enhancement architecture.

**[0086]** **FIG. 4*b*** is a block diagram showing stages of video coding according to the present invention.

**[0087]** **FIG. 4*c*** is a conventional decoder.

**[0088]** **FIG. 4*d*** is another conventional decoder.

**[0089]** **FIG. 4*e*** is another conventional decoder.

**[0090]** **FIG. 1*e*** is another conventional decoder.

**[0091]** **FIG. 4*f*** is another decoder.

**[0092]** **FIG. 5*a*** is a block diagram of a real-time process stage of an enhancement process according to the present invention.

**[0093]** **FIG. 5*b*** is a block diagram showing databases maintained by an encoder according to the present invention.

**[0094]** **FIG. 5*c*** is a block diagram showing look ahead stages of an enhancement encoder according to the present invention.

**[0095]** **FIG. 5*d*** is a block diagram showing a pre-classification stage according to the present invention.

**[0096]** **FIG. 5*e*** is a block diagram showing a circuit for authorizing figures according to the present invention.

**[0097]** **FIG. 5*h*** is a block diagram showing conventional DVD authorizing.

**[0098]** **FIG. 5*i*** is a block diagram showing storage prior to multiplexing a disc record.

**[0099]** **FIG. 5*j*** is a block diagram showing an alternate embodiment of generating an enhancement stream according to the present invention.

**[0100]** **FIG. 6*a*** is a block diagram showing stages within the prediction function according to the present invention.

**[0101]** **FIG. 6*b*** is a block diagram showing the generation of an enhanced picture.

**[0102]** **FIG. 6*c*** is a functional block diagram of a circuit for generating enhanced pictures according to the present invention.

**[0103]** **FIG. 6*d*** is a block diagram of a circuit for generating enhanced pictures according to the present invention.

**[0104]** **FIG. 7** shows syntax and semantic definitions of data elements according to the present invention.

**[0105]** **FIG. 7*a*** is a strip diagram according to the present invention.

**[0106]** **FIG. 7*b*** is a flow chart showing a procedure for passing a strip.

**[0107]** **FIG. 7*c*** is a flow diagram showing a block.

**[0108]** **FIG. 7*d*** is a block diagram showing codebook processing.

**[0109]** **FIG. 7*e*** is a diagram showing block delineation within a picture.

**[0110]** **FIG. 7*f*** is a diagram showing codebook selection by content region.

**[0111]** **FIG. 7*g*** is a diagram showing strip delineation according to region.

**[0112]** **FIG. 7*h*** is a video sequence comprising a group of dependently coded pictures.

**[0113]** **FIG. 8*a*** shows a conventional packetized elementary stream.

**[0114]** **FIG. 8*b*** shows a private stream type within a multiplex.

**[0115]** **FIG. 8*c*** shows conventional scenes and groups of pictures.

**[0116]** **FIG. 8*d*** shows a conventional relationship coded frame and display frame times.

**[0117]** **FIG. 8*e*** shows codebook application periods.

## OVERVIEW OF TOOLS

**[0118]** The PHD decoding process depicted in **FIG. 4*b*** has two fundamental stages of modem video coding. A first prediction phase **4130,1130** forms a first best estimate **4132, 1135** of the target picture **4152,1175**, using only the output state **4115,1115** of a base layer decoder **4110,1110** (and some minimal directives **4122**), followed by a prediction error phase comprising classification **4140,1120**, enhancement decode **4120,1150** and application **4150** of correction **1165** terms that improve the estimate.

**[0119]** The overall PHD enhancement scheme fits within the template of the classic spatial scalable enhancement architecture (**FIG. 4*a***). The respective base layer decoders **4020,4110** are principally the same. Both fundamental enhancement phases may operate concurrently in the receiver, and their respective output **4126,4032** added together at a later, third phase **4150**, where the combined signal **4152** is sent to display, and optionally stored **4160** for future reference **4172** in a frame buffer **4172**. In a simplified embodiment the enhanced reconstruction **4152** may be sent directly to display **4162** to minimize memory storage and latency.

**[0120]** As part of the estimation phase **4130**, the decoded base layer picture **4115** is first interpolated according to parameters **4122** to match the resolution of the reconstructed HD image **4152**. The interpolated image is a good first estimate of the target frame **4152**. Traditional interpolation filters are applied in the preferred embodiment during the interpolation process.

**[0121]** A first stage of the prediction error is to extract 4×4 blocks **1115** from the decoded base layer picture **(4115)** for classification analysis **4140**. In order to keep computational complexity to a minimum, the preferred embodiment does not classify the interpolated base layer picture **4132**, since the interpolated image nominally has four times the number pixels as the base layer image **4115**. The interpolated image

4132 is simply an enlarged version of the base layer image 4115, and inherently contains no additional information over the non-interpolated base layer image 4115.

[0122] The preferred embodiment employs vector quantization to generate correction terms, in the form of 8×8 blocks 4126. Each block, or codevector, within the codebook represents a small difference area between the interpolated predicted base image 4132 and the desired image 4152. The codebook comprising VQ difference blocks are stored in a look up table (LUT) 1160. The difference blocks are ideally re-used many times during the lifetime of the codebook.

[0123] Encoder

[0124] **FIG.** 5*c* denotes the time order of the multi-pass base 5220 and enhancement layer (5230, 5240) video encoding processes. Nominally, the base layer signal 5022 is first generated for at least the period that corresponds to the enhancement signal period coded in 5230. Alternative embodiments may jointly encode the base and enhancement layers, thus different orders, including concurrent order, between 5210 and 5230 are possible. The overall enhancement process has two stages: look-ahead 5230 (**FIG.** 5*d*) and real-time processes 5240 (exploded in **FIG.** 5*a*). The enhancement look-ahead period is nominally one scene, or access unit interval for which the codebook is generated and aligned. The iteration period may be one scene, GOP, access unit, approximate time interval such as five minutes, or entire program such as the full length of a movie. Only during the final iteration are the video bitstreams (5022, 5252) actually generated, multiplexed into the program stream 5262, and recorded onto DVD medium 5790. For similar optimization reasons, the final enhancement signal 5252 may also undergo several iterations. The multi-pass base layer encoding iterations offer an opportunity in which the PHD look-ahead process can operate without adding further delays or encoding passes over the existing passes of prior art DVD authoring.

[0125] **FIG.** 5*b* lists the databases maintained by the encoder 5110 look-ahead stages of **FIG.** 5*c*. The enhancement codebook 5342 (database 5140) is constructed by 5340 (described later) from training on blocks extracted from difference signal 5037 (database 5130). The codebook is later emitted 5232, packed 5250 with other enhancement sub-streams (5234, 5252) and data elements and finally multiplexed 5260 into the program stream 5262. In the preferred embodiment, the difference signal 5037 is generated just-in-time, on a block basis, from delayed pre-processed signal 5010 stored in buffer 5013 (database 5160). Likewise, the base layer signal 5032 (database 5120) is generated just in time from decoded SD frames (database 5150). Alternative embodiments may generate any combination of the signals that contribute to the enhancement stream encoding process, either in advance (delayed until needed by buffers), or just-in-time.

[0126] The first two pre-classification stages 5310, 5320, described later in this document, produce two side information arrays (or enhancement streams) 5325 and 5315 (database 5180) that are later multiplexed, along with the codebook, into the packed enhancement stream 5252. The results of the third pre-classification stage 5332 of **FIG.** 5*d* may be temporarily maintained in encoder system memory, but are used only for codebook training.

[0127] Although original HD frames (signal 5007) are in the preferred embodiment are passed only to the pre-processor 5010, further embodiments may keep the frames (database 5170) for multi-pass analysis in the classification or codebook training phases.

[0128] Run-time operations 5240, whose stages are detailed in **FIG.** 5*a*, can be generally categorized as those enhancement stages that produce packed bitstream elements for each coded enhancement picture. The enhancement data may be buffered 5820 or generated as the final DVD program stream is written to storage medium 5790 master file. Buffering 5820 allows the enhancement stream to have variable delays to prevent overflow in the system stream multiplexer 5260. Enhancement may be generated in step with the base layer 5020 encoder at granularities of a blocks, macroblocks, macroblock rows and slices, pictures, group of pictures, sequences, scenes or access units. An alternate embodiment (**FIG.** 5*j*) is to generate the enhancement stream 5252 after the base layer signal 5022 has been created for the entire program, as would be the case if the enhancement is added to a pre-existing DVD title.

[0129] A second alternate embodiment is to generate the base and enhancement layers jointly. A multi-pass DVD authoring strategy would entail several iterations of each enhancement look-ahead process, while the joint base and enhancement rate controllers attempt to optimize base and enhancement layer quality.

[0130] For best coding efficiency, the applied codebook and enhancement stream are generated after the scene, GOP (Group of Pictures), or other interval of access unit has been encoded for the base layer. The delay between base layer and enhancement layer steps is realized by buffers 5013 and 5023.

[0131] The pre-processor 5010 first filters the original high-definition signal 5007 to eliminate information which exceeds the desired rendering limit of the PHD enhancement process, or patterns which are difficult to represent with PHD. The outcome 5012 of the pre-processor represents the desirable quality target of the end PHD process. Film grain and other artifacts of the HD source signal 5007 are removed at this stage.

[0132] The SD source signal 5017 is derived from the pre-processed HD signal 5012 by a format conversion stage 5015 comprising low-pass filters and decimators. The SD signal 5017 serves as source input for MPEG-2 encoding 5020.

[0133] MPEG-2 encoder 5020 produces bitstream 5022, that after delay 5023, is multiplexed as a separate elementary stream 5024 in the program stream multiplexer 5280.

[0134] The SD signal 5027 reconstructed by MPEG-2 decoder 5025 from delayed encoded SD bitstream 5024 is interpolated 5030 to serve as the prediction for the target HD signal 5014.

[0135] The prediction engine 5030 may also employ previously enhanced frames 5072 to form a better estimate 5032, but nominally scales each picture from SD to HD dimensions.

[0136] The difference signal 5037 derived from the subtraction 5035 of the predicted signal 5032 from the HD target signal 5014 serves as both a training signal and

enhancement source signal for the PHD encoding process **5050**. Both source signals require the corresponding signal components generation within the PHD encode process **5050** and enhancement coding

[0137] The classifier **5040** analyzes the decoded SD signal **5027** to select a class **5047** for each signal portion, or block, to be enhanced by the PHD encoding process **5050**. The encoded enhancement signal **5052** is decoded by the PHD decoder **5060**, which in the encoder system can be realized as a look up table alone (**5061**) since the indices exist in pre-VLC (Variable Length Coding) encoded form within the encoder. The decoded enhancement signal **5062** is added by **5065** to the predicted HD signal **5032** to produce the reconstructed HD signal **5067**. The goal of the PHD encoder is to achieve a reconstruction **5067** that is close to the quality of the target HD signal **5014**.

[0138] The reconstructed HD signal **5067** may be stored and delayed in a frame buffer **5070** to assist the interpolation stage **5030**.

[0139] The encoded PHD enhancement signal **5052** is multiplexed **5260** within the DVD program stream as an elementary stream with the base layer video elementary stream **5024**.

[0140] Some stages of the run-time operations are common to both the encoder and decoder. The encoder explicitly models decoder behavior when a decoded signal is recycled to serve as a basis for prediction **5072** in future signals, or when the decoder performs some estimation work **5040** of its own. For similar reasons, the MPEG-2 encoder **5020** models the behavior of the MPEG-2 decoder **5025**.

[0141] Pre-Processor (5010)

[0142] The primary responsibility of the pre-processor **5010** is to perform format conversion that maps the master source signal **5007** to the sample lattice of the HD target signal **5014**.

[0143] The most common source format for HD authoring is SMPTE 274M, with 1920 luminance samples per line, and 1080 active lines per frame. In order to maintain a simple 2:1 relationship between the base and enhancement layers, and to set a realistic enhancement target, the preferred enhancement HD coding lattice is twice the horizontal and vertical dimensions of the coded base layer lattice. For "NTSC" DVD's, this is 1440x960 and 1408x960 for respective 720x480 and 704x480 base layer dimensions. For "PAL" DVD's with 576 active vertical lines, the enhancement dimensions are 1440x1152 and 1408x1152 respectively. The base layer will assumed to be 720x480 for purposes of this description, although the enhancement process is applicable to any base and enhancement dimension, and ratio.

[0144] A skilled engineer can chose from many image scaling designs, including well known poly-phase FIR filters, to convert the first 1920x1080 frame lattice of **5012** to the second 1440x960 lattice of **5017**. Another possible formats for either or both of the input **5012** and output **5017** sides is SMPTE 296M, with 1280x960 image dimensions. A corresponding format conversion stage **1482** in the decoder maps the PHD coded dimensions to the separate requirements of the display device connected to display signal **1482**. Common display formats include SMPTE 274M (1920x1080x30i) and SMPTE 296M (1280x720x60p).

[0145] General format conversion pre-processing essentially places the target signal in the proper framework for enhancement coding. The goal of pre-processing is to produce a signal that can be efficiently represented by the enhancement coding process, and assists the enhancement coder to distribute bits on more visibly important areas of the picture. Several filters are employed for the multiple goals of pre-processing.

[0146] A band-pass filter eliminates spatial frequencies exceeding a user or automatically derived target content detail level. The band-pass filter can be integrated with the format conversion scaling filters. The format scaling algorithm reduces the 1920x1080 HD master format to the 1440x960 coding format, but additional band-pass filtering smoothes the content detail to effectively lower resolutions, for example, 1000x700.

[0147] Adaptive filtering eliminates patterns that are visually insignificant, yet would incur a bit cost in latter encoding stages if left unmodified by the pre-processor. Patterns include film grain; film specs such as dirt, hair, lint, dust;

[0148] A classic pattern and most common impediment to efficient coding is signal noise. Removal of noise will generally produce a cleaner picture, with a lower coded bit rate. For the PHD enhancement process, noise removal will reduce instances of codebook vectors that would otherwise be wasted on signal components chiefly differentiated by noise. Typical noise filters include 2D median, and temporal motion compensated IIR and FIR filters.

[0149] Downsample (5015)

[0150] The base layer bitstream complies with MPEG-2 Main Profile @ Main Level video sequence size parameters fixed by the DVD specification. Although MPEG-2 Main Profile @ Main Level can prescribe an unlimited number of image size combinations, the DVD specification limits the MPEG-2 coding parameters to four sizes (720x480, 704x480, 720x576, and 704x576), among which the DVD author can select. The DVD MPEG-1 formats (352x240 and 352x288) are not described here, but are applicable to the invention. The HD target sample lattice **5012** is decimated **5015** to the operational lattice **5017** of the MPEG-2 **5020**. Downsampling **5015** may be bypassed if the encoder **5020** is able to operate directly upon HD formats, for example, and is able to perform any necessary conversion to the DVD base layer video format. In prior art, downsampling **5015** will execute master format conversion, such 24p HD (SMPTE RP 211-2000) to the SD format encoded by **5020**.

[0151] Downsampling may be performed with a number of decimation algorithms. A multi-tap polyphase FIR filter is a choice.

[0152] MPEG-2 Encoder (5020)

[0153] The MPEG-2 encoder **5020** nominally performs as prior art encoders for DVD authoring. Although the invention can work with no changes to the base layer encoder **5020**, improvements to the overall reconstructed enhancement layer video can be realized through some modification of the base layer encoding process. In general, any operation in the base layer that can be manipulated to improve quality or efficiency in the enhancement layer is susceptible to coordination with the enhancement process. In particular, operation of the DCT coefficient quantizer mechanisms

quant_code and quantization_weighting_matrix can be controlled to maintain consistent enhanced picture quality. In some combinations of base and enhancement data, this would be more efficient than applying additional bits to the corresponding area in the enhancement layer. In an advanced design, the rate control stage of the encoder **5020** could have dual base and enhancement layer rate-distortion optimization.

[0154] Improved motion vectors coding in the base layer may benefit modes of the enhanced prediction stage **5030** that employ motion vectors extracted from the base layer signal **5022** to produce interpolated predicted frames (a feature of an alternate embodiment described later in this specification). Motion vector construction is directly operated by rate-distortion optimization with feedback from both the base and enhancement reconstruction.

[0155] The encoder may also need to throttle back the bitrate to ensure the combination of enhance and base bitstreams do not exceed DVD buffer capacity.

[0156] Prediction (5030)

[0157] The prediction scheme forms a best estimate of the target signal by maximizing use of previously decoded data, and thereby minimizing the amount of information needed for signaling prediction error. For the application of picture resolution and detail enhancement, a good predictor is the set of image interpolation algorithms used in scaling pictures from one resolution, such as an intermediate or coded format, to a higher resolution display format. These scaling algorithms are designed to provide a plausible approximation of signal content sampled at higher resolution given the limited information available in the source lower resolution picture.

[0158] Overall, the base layer decoded image **6110** extracted from signal **5027** is scaled by a ratio of 2:1 from input dimensions 720×480 to an output dimension of 1440×960 of the signal **5032** to match the lattice of the target **5014** and enhanced images **5067** so that the predicted signal **5032** image **6120** may be directly subtracted **5035** from the target signal **5014**, and directly added **5065, 6130** to the enhancement difference signal **5062** image **6140** to produce the enhanced picture **6150**. Other ratios and image sizes are applicable. In some picture areas or blocks, the predicted signal **5032** is sufficient in quality to the target signal **5014** that no additional information **5052** need be coded.

[0159] The order of the stages within the prediction **5030** function of the preferred embodiment is depicted in **FIG. 6***a*. Other orders are possible, but the preferred order is chosen as a balance between implementation complexity and performance, and for dependencies with the base layer bitstream such as the de-blocking stage's use of quantizer step sizes. Starting with the base frame **6010, 6110** extracted from signal **5027**, a de-blocking filter **6020** is applied to reduce coding artifacts present in the base layer. Although good coding generally yields few artifacts, they may become more visible or amplified as a result of the scaling process **6030**, or plainly more visible on a higher definition screen. De-blocking reduces unwanted patterns sometimes unavoidably introduced by the MPEG-2 base layer encoding process **5020**.

[0160] The de-blocking filter of ITU-T H.263 Annex J is adapted to **6020**. Some stages of the Annex J filter require modifications in order to fit the invention. For example, the de-blocking filter is performed as a post-processing stage after the image has been decoded, not as part of the motion compensated reconstruction loop of the base layer decoder. The quantization step function is remapped from the H.263 to the steps of the MPEG-2 quantizer. The strength of the de-blocking filter is further regulated by a global control parameter transmitted with each enhanced PHD picture. The PHD encoder sets the global parameter to weight the Annex J STRENGTH constant according to analysis of the decoded picture quality. Since the quantizer scale factor is not always an indication of picture quality or coding artifacts, the PHD encoder aims to use the global parameter to set the STRENGTH value to minimal for pictures with excellent quality, thus de-blocking is effectively turned off when it is not needed or would do unnecessary alterations to the picture.

[0161] A poly-phase cubic interpolation filter **6030** derives a 1440×960 image **6035** from the de-blocked standard definition 720×480 image **6025**.

[0162] Post-filtering **6040** optionally performs de-blocking on the scaled image **6035** rather than the base layer image **6015**.

[0163] In an alternative embodiment (**FIG. 6***c* functional blocks and **FIG. 6***d* data blocks), a subset of pictures within a sequence or GOP are alternatively predicted from a combination of previously decoded base layer and enhanced pictures **6320,6322** stored in frame buffer **6225**—a subset of frame buffer **5070**. This variation of a predicted enhancement picture is henceforth referred to as a temporally predicted enhancement picture (TPEP) **6345**. TPEP resembles the B-frame or "bi-directionally" predicted frames since they borrow information from previously decoded frames that in display order are both future and past. The difference enhancement **6320, 6322** from previously decoded pictures is re-applied to the current picture **6315** as a good estimate of the enhancement difference **6140** that would be otherwise transmitted as enhancement data in non-TPEP pictures. TPEP is a tool for reducing the overall or average bitrate of the enhancement layer since data is not often coded for TPEP blocks. If difference mode is enabled in the header of TPEP pictures, a 1-bit flag prefixes each TPEP block indicating whether difference information will be transmitted for the block. TPEP pictures are enabled when the corresponding base layer picture is a B picture; the scaled motion information **6235** from the base layer picture instructs the MCP **6235** to create the prediction surface **6325** that is combined **6340** with the interpolated base frame **6315**.

[0164] Classification

[0165] While Standard Definition (SD) and High Definition (HD) images captured of the same scene differ superficially by the density and size of their respective sample lattices (1440×960 vs. 720×480), they may substantively differ in content, in particular when analyzed in the frequency domain. Generally, a hierarchical relationship should exist in that the information in the SD image is a subset of the HD image, such that the SD image may be derived from the HD image through operations such as filtering and sub-sampling. (Eq.1)

$$SD = \text{sub-sample}(HD) \quad \text{(Eq. 1)}$$

[0166] In the spatial domain, an HD image can be represented as the sum of a first base image (B) and a second difference (D) image:

$$B=\text{sub-sample (HD)}$$

$$D=\text{HD}-B \quad \text{(Eq. 2)}$$

$$\text{HD}'=B'+D \quad \text{(Eq. 3)}$$

[0167] In this example, the difference image (D) contains the high frequency components that distinguish the HD image from the SD image, while the base image (B) contains the remaining low frequency information. When the base image (B) by itself can serve as the SD image, the difference image (D) could then be formulated to contain the set of information that is present only in the HD image, not the SD image.

[0168] Further, the SD image can be sampled at a reduced resolution, with a smaller lattice (such as 720×480), sufficient to contain the lower half of the frequency spectrum, and later scaled (SD') to match the sample lattice (e.g. 1440×960) of the HDTV image where it may be easily recombined in the spatial domain with the difference image (D) to produce the reconstructed HD image (HD').

[0169] While the lower frequencies are significantly more important than high frequencies in terms of perceptible contribution to the overall image (HD'), the high frequency information is still needed to establish the "look and feel" of an HD image.

[0170] Although the difference image may be expected to contain up to three times more information than the base image, not all portions of the difference image contribute equally to the overall perceptible quality of the final reconstructed HD image. The essential information in (D) needed to emulate the look and feel of the HD image may in fact be a small subset of D, in particular concentrated along edges and areas of texture, and may be further approximated very coarsely. This concept is essentially supported by the practice in the block coding methods of JPEG and MPEG where high frequency DCT coefficients are more coarsely quantized than low frequency DCT coefficients.

[0171] The MPEG coding tools are not optimized for coding these essential difference areas efficiently at extremely low bit-rates (or in other words, high compression factors). MPEG is tuned towards visual approximation of an image with a balance of detail and generic content at appropriately matched resolutions. For example, the luminance samples of a typical still frame will be represented as an MPEG intra-frame (I) in approximately one fourth the rate of the "non-coded" PCM frame, and the average predicted frame (P,B) only one fifteenth the size of the PCM frame.

[0172] The classifier stage of the invention serves as a key tool for identifying those areas of the picture of greater subjective importance, so that enhancement coding may be emphasized there. At the same time, the process also objectively places emphasis on those areas where the difference energy is greater, such as edges.

[0173] Strong horizontal, vertical, and diagonal edges, for example, can be identified at lower resolutions, such as the SD base layer. It is possible to identify within the SD image areas that should result in a combination of high frequency and high perceptible patterns in the HD image. Unfortunately, sufficient clues in the base image are not accessible to accurately estimate the actual difference information for those areas, although reasonable guesses bounded by constraints imprinted in the base layer are possible, and have been developed by various prior "sub-pixel" developments. To meet real-time implementation constraints, prior art interpolation schemes would generate "synthetic highs" through contrast enhancement or sharpening filters. The most common algorithm for interpolating image is a filter that convolves the lower resolution samples with a curve that models the distribution of energy in the higher resolution sample lattice, such as the sinc( ) function.

[0174] Superficially sharp, high resolution images restored by synthetically means from low resolution images often looks contrived or artificial byproduct, and quality gains may be inconsistent.

[0175] Accurate identification of picture areas is possible with knowledge of the original HD image, but such an image is available only to the encoder residing at the transmitter side. Enhancement information can be explicitly transmitted with this knowledge to guide the HD reconstruction process, and thus produce more natural looking "highs". However enhancement data can easily lead to a significant bit rate increase over the base layer data.

[0176] The more accurate the highs can be estimated by the receiver, the less enhancement information is needed to improve the reconstructed HD signal to a given quality level. A particular tool useful for minimizing the volume of enhancement information is classification.

[0177] Classification can be used to partially predict the enhancement layer and/or prioritize those areas that need to be enhanced. Classification also permits different coding tools to be used on different classes of picture data. For example, in flat areas the SD to HD interpolation algorithm may dither, while pixels determined to belong to an edge class may benefit from directional filtering and enhancement data.

[0178] As appropriate for the overall enhancement technique, classification can be accomplished in the frequency or spatial domains. A classifier is also characterized by the granularity of the classified result (such as on a per pixel or block basis), and by the window of support for each granule.

[0179] The window of the classifier is the size of the support area used in the classification analysis. For example, to determine the class of a single target pixel, the surrounding 5×5 area may be measured along with the target pixel in order to accurately measure its gradient.

[0180] Familiar to video compression, a good balance between implementation complexity, bitrate, and quality can be achieved with block-based coding. The negative tradeoff is manifested by inaccuracies that result at block edges and the other blocking artifacts.

[0181] The preferred PHD classification scheme employs block-based frequency and spatial domain operators at a granularity of 4×4 pixels with respect to the base layer, and 8×8 pixels with respect to the HD image. Local image geometry (flat, edge, etc.) is first determined through a series of comparisons of measurements derived from frequency coefficients of a 4×4 DCT taken on a non-overlapping block within in the base image. Overlapping is also possible, but

not implemented in the preferred embodiment. The small 4×4 block size has many of the desired properties of a local spatial domain operation, but with greater regularity and reduced complexity compared to both per-pixel granular operations, and generally most known effective all-spatial domain operations.

[0182] Calculating Classification Components

[0183] **FIGS. 3***b* and **3***d* provide the fundamental stages of the preferred classifier embodiment that are common to both the encoder and decoder. **FIG. 3***d* discloses the classifier component calculations **3130** of **FIG. 3***b*.

[0184] Blocking

[0185] Blocks of data are extracted from the input frame **3100** in the processing order of the enhancement decoder. The preferred processor order is raster, from left to right and top to bottom of the picture, with non-overlapping blocks. Alternate embodiments may overlap blocks in order to improve classification accuracy. For example, a 3×3 target block may be processed from a 4×4 input block. In the 3×3 within 4×4 block example, the overlap areas would comprise a single row and column of extra pixels. Each successive 3×3 picture area would then be processed from a 4×4 block with a unique combination of samples formed from the base picture. The 4×4 input block would step three pixels for each advance in either or both the x and y directions. A new set of classification parameters would be derived for each 3×3 picture area. Other overlaps are possible, but in general, the overlap and target blocks may be arbitrarily shaped as long as the base and enhancement layers are aligned.

[0186] DCT

[0187] In the preferred embodiment, the DCT-II algorithm is applied in the 4×4 DCT **3312** to produce the coefficients **3314** whose combinations are used as feature component measurements **3332** for the decision stage **3140**. Variations include the DCT-I and DCT-III, non-DCT algorithms, and pseudo-DCT algorithms such as those experimented with by the ITU-T H.264 study group. Generally, any transform which produces coefficients useful in the classification of a picture area can substitute for the preferred block DCT, however adjustments to the ratio calculations in **3130** and decision tree **3140** may be necessary to account for the different characteristics of each transforms unique coefficient sets.

[0188] The 8-bit precision of the transform coefficients and 16-bit intermediate pipeline stages are sufficient to support the expansion of data in the transform size and the accuracy needed to discriminate one class from another. The preferred transform is designed to operate within the 16-bit SIMD arithmetic limitations of the Intel MMX architecture which serves as an exemplary platform for PHD DVD authoring.

[0189] Spatial analysis

[0190] The Weber function provides a more accurate measurement of picture area flatness than a single combination of DCT coefficients.

[0191] The Weber component **3322** calculated in **3320** follows the formula summarized as:

[0192] compute difference between max value of block and average block value if the difference/average<=0.03, then it is flat (isFlag=1), else isFlag=0.

[0193] Frequency Analysis

[0194] Component generator **3330** takes measurements **3132** conducted on the 4×4 blocks and produces decision variables **3332, 3132** used in the decision process **3140** to create classification terms **3142**. The block measurements **3132** comprise both frequency measurements **3314** (in the preferred embodiment realized by the 4×4 DCT transform **3312**) and spatial domain measurements **3322** (in the preferred embodiment realized by a flatness operator **3320**).

[0195] Input blocks **3310, 3122** formatted from the base layer reconstructed image **3100** are transformed via the 4×4 DCT **3312**, producing coefficients **3314**. The component generator stage **3332** takes sets of coefficients **3314** shown in **FIG. 3***e*, and squares and sums coefficients within each set to produce class components **3332**, P1 through P7. Each set of DCT coefficients, and its resulting measurement term (P1 . . . P7), represents the identifying characteristic of a geometric shape such as an edge, texture, flat area. =p The seven 4×4 DCT coefficient templates in **FIG. 3***e* shows increasing horizontal frequency is along the U-axis with set of indices {0, 1, 2, 3 }, and increasing vertical frequency along the V-axis with indices {A, B, C, D}.

[0196] Each of the components P1 . . . P7 represent the following geometry features: P1—horizontal edges, P2—horizontal texture, P3—vertical edges, P4—vertical texture, P5—diagonal edges, P6—texture, and P7—energy/variance of the block.

[0197] (P1) diag=B1*B1+C2*C2+D3*D3

[0198] (P2) inf0=B0*B0+C0*C0+D0*D0+C1*C1+D1*D1+D2*D2

[0199] (P3) inf1=B0*B0+C0*C0+D0*D0

[0200] (P4) sup0=A1*A1+A2*A2+A3*A3+B2*B2+B3*B3+C3*C3

[0201] (P5) sup1=A1*A1+A2*A2+A3*A3

[0202] (P6) text=C2*C2+C3*C3+D2*D2+D3*D3

[0203] (P7) tot=diag+sup0+inf0

[0204] Ratios:

[0205] From the seven component measures (P1 . . . P7), eight ratios (R0 . . . R7) are derived that are used in the decision process **3140** to select the class for each block.

[0206] R0=diag/tot

[0207] R1=sup0/(sup0+inf0)

[0208] R2=sup1/sup0

[0209] R3=inf0/(sup0+inf0)

[0210] R4=inf1/inf0

[0211] R5=text/(sup0+inf0)

[0212] R6=sup1/(sup0+inf0)

[0213] R7=inf1/(sup0+inf0)

[0214] Pre-Calculated Ranges

[0215] In order to improve accuracy of the codebook and run-time classification passes, two pre-classification passes **5310, 5320, 5330** are made through the decoded base layer signal **5027, 5305**, to measure the statistics of classification

components. Specifically, thresholds **5317** and energy ranges **5327** are produced in the first and second passes respectively. The third classification pass **5330** selects the class for each training block **5332** used in codebook generation stage **5340**. The codebook is trained on the decoded base layer signal; the results of the third pre-classification stage therefore **5332** model (sans IDCT drift error) the run-time classifier **5040** results of downstream decoder classifier.

[0216] Ratios R0 . . . R7 are calculated in the classification stage as above, and then compared to pre-determined thresholds to establish 17 energy ranges **5327**.

[0217] Ranges and thresholds (shown collectively as side information **5234**) are maintained in memory **5180** for later application in the class decision stage **3140**. To save computation time, and spare the decoder from having to add significant latency, the encoder packs the ranges and thresholds into the PHD stream **5252**, where on the receiver side, they are later parsed and integrated into the state machine **3620** by the PHD decoder during each codebook update.

[0218] To improve accuracy of classification, the components used in the classification decision process are adaptively quantized according training block statistics. The quantized levels are indicated by thresholds **5315** which are calculated from an equi-probable partitioning of histograms measured during the first pre-classification training pass **5310**.

[0219] Pass 1. generate adaptive quantization thresholds:

[0220] For each training block.

[0221] if ((R1>0.60) && (R2<=0.90)) hist_add(hist1, R1);

[0222] else if ((R1>0.60) && (R2>0.90)) hist_add( hist2, R1);

[0223] else if ((R3>0.60) && (R4<=0.90)) hist_add( hist3, R3);

[0224] else if ((R3>0.60) && (R4 >0.90)) hist_add (hist4, R3);

[0225] Hist_add( arg1, arg2) updates respective histogram (indicated by arg1) with the data point arg2. Each histogram is allocated to track a range of values divided into a specified number of partitions. Each update of arg2 will increment the corresponding partition identified by arg2 by one count.

[0226] At the end of the training sequence, hist_convg(arg1, arg2, arg3, arg4) partitions thresholds **5315** (arg3) into arg4 number of equi-probable partitions according to the statistics stored in the respective histogram arg1:

[0227] At the end of the training session.

[0228] hist_convg( hist1, hcenters, thresh1, 2);

[0229] hist_convg( hist2, hcenters, thresh2, 5);

[0230] hist_convg( hist3, hcenters, thresh3, 2);

[0231] hist_convg( hist4, hcenters, thresh4, 5);

[0232] The second parameter, arg2, of Hist_conv( ) provides additional statistics including the average and standard deviation squared of each partition.

[0233] Pass 2, measure energy:

[0234] Note: isFlat is the result of the Weber calculation **3320**.

```
if (isFlat)
        idx = 0;
else
{
    if (R0 >= 0.55)
        idx = 1;
    else
    {
        if ((R1 > 0.60) && (R2 <= 0.90))
        {
            if (R1 < thresh1[0])
              idx =2;
            else
              idx = 3;
        }
        else if ((R1 > 0.60) && (R2 > 0.90))
        {
            if (R1 < thresh2[0])
              idx = 4;
            else if (R1 < thresh2[1])
              idx = 5;
            else if (R1 < thresh2[2])
              idx = 6;
            else if (R1 < thresh2[3])
              idx = 7;
            else
              idx = 8;
        }
        else if ((R3 > 0.60) && (R4 <= 0.90))
        {
            if (R3 < thresh3[0])
              idx = 9;
            else
              idx = 10;
        }
        else if ((R3 > 0.60) && (R4 > 0.90))
        {
            if (R3 < thresh4[0])
              idx = 11;
            else if (R3 < thresh4[1])
              idx = 12;
            else if (R3 < thresh4[2])
              idx = 13;
            else if (R3 < thresh4[3])
              idx = 14;
            else
              idx = 15;
        }
        else
            idx = 16;
t[idx][count[idx]] = Etot;
count[idx] = count[idx] + 1;
min_energy_class [idx] =
     MYMIN ( min_energy_class[idx], Etot );
max_energy_class [idx] =
     MYMAX ( max_energy_class[idx], Etot );
```

[0235] At the end of the second pre-classification pass **5320** of the training sequence, the statistics in temporary variable arrays t[] and count[] are used to calculate 17 energy_range[]**5325** constants used in the classification stage.

```
for (i = 0; i > 17; i++)
{
    median(count[i],&t[i][0],&median_val);
    energy_range[i] = median_val;
}
```

[0236] Determining Class by Decision Tree

[0237] To arrive at a specific class, the classifier uses the component measurements produced in **3510**, **3330**, to descend a decision tree, comparing class components **3332** and pre-calculated ranges (**3102**, **5180**, **5240**, **5234**, **5315**). The generic cyclical flow of the classification process is given in **FIG.** 3*f*. Comparisons are made **3520** until a state process indicates that a class has been arrived at **3530**. With the binary decision branch process depicted, the number of iterations should be approximately the logarithm of the number of available classes. Means of implementing the decision tree include procedural code (nested if statements) given below, and parallel flow-graph testing (not shown).

[0238] A state machine realization of the decision tree is given in flowchart **FIG.** 3*g*. The state machine is expected to be the easiest State parameters table **3620** is indexed by variable L, initialized to zero **3610**. The resulting state parameters **3621** include branch positive address L1, branch negative address L2, classification component identifiers p1 and p2, multiplier constant k, offset T. and termination bits e1 and e2.

[0239] Component identifiers p1 and p2 select which classification ratios in the set P1 . . . P7 are to be compared in **3640**. The values for p1 and p2 are selected **3630** from the class component register array cc and compared as a and b in formula **3640**. The branch addresses L1 are the next location in the state code **3620** that the state program reaches if the comparison in **3640** is positive, and L2 is the location if the comparison is negative. If either or both of the comparison results indicate a terminal condition, that is a terminal node with a specific class is finally reached, then either or both terminal state bits e1, e2 will be set to '1' potentially causing the loop to exit Y at **3650**. In a terminal cases (where E==1), state variables L1 and L2 encode the class index **3632** which forms part of the state **3142** in **FIG.** 3*b* needed to perform, at least, the LUT **3150**.

[0240] A procedural example of the decision tree is below. Energy_clase.

```
if (isFlat)
    energy_class[i] = 0;
else
{
    if (R0 >= 0.55) // diagional
    {
        if (Etot < energy_range[1])
            {
                energy_class[i] = 1;
            }
            else
            {
                energy_class[i] = 2;
            }
    }
    else
    {
        if ((R1 > 0.60) && (R2 <= 0.90))
        {
            if (R1 < thresh1[0]) // vert_text_0
            {
            if (Etot < energy_range[2])
                energy_class[i] = 3;
            else
                energy_class[i] = 4;
            }
```

-continued

```
            else // vert_text_1
            {
            if (Etot < energy_range[3]) // vert_text
                energy_class[i] = 5;
            else
                energy_class[i] = 6;
            }
}
else if ((R1 > 0.60) && (R2 > 0.90))
{
    if (R1 < thresh2[0]) // count_vert_0
    {
        if (Etot < energy_range[4])
            energy_class[i] = 7;
        else
            energy_class[i] = 8;
}
else if (R1 < thresh2[1]) // vert_1
{
    if (Etot < energy_range[5])
        energy_class[i] = 9;
    else
        energy_class[i] = 10;
}
else if (R1 < thresh2[2]) // vert_2
{
    if (Etot < energy_range[6])
        energy_class[i] = 11;
    else
        energy_class[i] = 12;
}
else if (R1 < thresh2[3]) // vert_3
{
if (Etot < energy_range[7])
    energy_class[i] = 13;
else
    energy_class[i] = 14;
}
else // vert_4
{
    if (Etot < energy_range[8])
        energy_class[i] = 15;
    else
        energy_class[i] = 16;
}
else if ((R3 > 0.60) && (R4 <= 0.90))
{
    if (R3 < thresh3[0]) // text_0
    {
        if (Etot < energy_range[9])
            energy_class[i] = 17;
        else
        energy_class[i] = 18;
        }
        else // horz_text_1
        {
        if (Etot < energy_range[10])
            energy_class[i] = 19;
        else
            energy_class[i] = 20;
        }
}
else if ((R3 > 0.60) && (R4 > 0.90))
{
    if (R3 < thresh4[0]) // horz_0
    {
        if (Etot < energy_range[11])
            energy_class[i] = 21;
```

-continued

```
            else
              energy_class[i] = 22;
            }
            else if (R3 < thresh4[1]) // horz_1
            {
            if (Etot < energy_range[12])
              energy_class[i] = 23;
            else
              energy_class[i] = 24;
            }
            else if (R3 < thresh4[2]) // horz_2
            {
            if (Etot < energy_range[13])
              energy_class[i] = 25;
             else
                energy_class[i] = 26;
            }
            else if (R3 < thresh4[3]) // horz_3
            {
            if (Etot < energy_range[14])
              energy_class[i] = 27;
            else
                energy_class[i] = 28;
            }
            else
            {
             if (Etot < energy_range[15]) // horz_4
              energy_class[i] = 29;
             else
              energy_class[i] = 30;
                count_++;
      }
}
else // ((R5 < 0.35) && (R6 < 0.65) && (R7 < 0.65))
{ // text_0
      if (Etot < energy_range[16])
        energy_class[i] = 31;
      else
        energy_class[i] = 32;
      }
```

[0241] Entire scenes, or individual pictures often do not contain significant detail in the original high-definition format signal beyond the detail that would be prescribed in any standard definition derivative of the high-definition signal. In such cases when there is insufficient difference between the high definition original signal **5012** and predictive signal **5032**, it more efficient to turn off enhancement block coding, while predictive interpolation continues to operate under both conditions in one mode or another.

[0242] To determine whether enhancement blocks should be sent for an area (encapsulated as a stripe), picture, or scene, the selective enhancement analyzer **5420** estimates the perceptivity of the difference signal **5037** for each block prior to both the VQ codebook training and run-time coding phases. Although many models exist for perceptivity, the simple energy formula calculated as the square of all N elements within the block serves as a reasonable approximation. The preferred embodiment applies the following formula:

$$e = \sum_{i=0}^{N-1} (\text{block } [i])^2$$

[0243] Three control parameters **5422** regulate the selection algorithm in **5420**. The first user control parameter, enerqy_threshold, sets the level of energy for a block to meet in order to be selected for enhancement by the encoder. Since the measurement is made on the difference signal **5037**, only the encoder can make such a judgment, although special cases such as flat areas (described earlier) that do not have associated indices are determined by the receiver through measurements on the base layer signal.

[0244] User control parameter stripe_block_ratio_threshold sets the minimum ratio of selected blocks within a stripe that must meet the perceptivity criteria in order for the slice to be coded. User control parameter block_max sets the level in which, regardless of the ratio of selected enhancement blocks, the stripe would be coded. This accounts for isolated but visually significant blocks.

[0245] Stripe headers include a 3-bit modulo index strip_counter so that the decoder can distinguish between non-coded gaps in the enhancement picture and stripes that have been lost to channel loss such as dropped or corrupted packets.

[0246] Blocks that do not meet the enhancement threshold are not applied during the VQ training process.

[0247] The is_picture_enhanced variable in the picture header signals whether enhancement blocks are present for the current picture. For finger granular control, the is_strip_enhanced flag in the strip header can turn enhancement blocks on or off for all blocks within a strip( ). In many cases, only a small subset of the picture has sufficient detail to merit enhancement, usually those areas that the camera had in focus. In such cases, the encoder can adapt the strip( ) structure to encapsulate only those detail areas, and leave the rest of the picture without strip( ) coverage. The x-y position indicators within the strip( ) header allow the strip( ) to be positioned anywhere within the picture.

[0248] PHD Run-Time Encoding (**5050**)

[0249] Enhancement data **5052** is generated for those blocks whose class has associated enhancement blocks **5062**. Of the thirty three classes, class 0, the category for flat areas, requires no transmission of indices. The statistical expectation is that at least one in three blocks will be classified as flat, and for some scenes, flat blocks will constitute a majority of blocks. Thus the bitrate savings can be substantial by not transmitting enhancement block indices for areas that do not sufficiently benefit from enhancement. Since the encoder and decoder have an identical understanding the enhancement syntax and semantics, the decoder parser does not expect indices for non-coded enhancement blocks.

[0250] For those classes with associated enhancement data, the VLC index is packed within the enhancement bitstream **5262** along with other enhancement elements. The combination of class and the VLC index are all that is needed to perform an enhancement pattern lookup **5060**, where a difference block is generated **5062** and added **5065** to the corresponding predicted-interpolated block **5032**. The same lookup procedure is performed in the receiver.

[0251] Small discrepancies in the reconstructed enhanced signal **5067** may exist due to difference among standard-compliant MPEG video reconstructions **5024**. No one model of the decoder **5025** applies universally. Drift free reconstruction is possible only if the IDCT in the encoder is

matched to the IDCT in the receiver. The difference signal, or drift, between the model decoder **5025** and the actual downstream decoder originates due to round-off errors in the integer approximation of the standard-defined floating point IDCT algorithm. The drift should be limited to an occasional least significant bit difference, but in pathological cases designed to accumulate worst case patterns, drift has been known to build to visible artifacts. Consequentially, drift can cause discrepancies between the encoder model classifier result **5047** and classification result **4142** in the downstream decoder. With proper threshold design, these discrepancy cases are rare and detectable through the class_checksum mechanism in the header of each strip( ). When class_checksum and the receiver calculated checksum differ, enhancement is not applied for those blocks for which the checksum applies. The specific class_checksum element applies to all blocks contained within the strip( ).

[0252] The preferred embodiment applies the well known CRC-32 algorithm to generate the bitstream checksum class_checksum and receiver checksum to which it is compared. Other hash algorithms could be applied, but CRC-32 circuitry is common in existing receivers with MPEG-2 video decoders.

[0253] Entropy Coding

[0254] The JPEG-2000 arithmetic coder is utilized by the invention for both codebook and enhancement block index transmission.

[0255] New codebooks are transmitted as raw samples. One codebook is sent for each class that has specified transmitted indices. For classes that do not have codevectors, the size_of_class variable (**FIG. 7**) is set to zero. The order of the codevectors within each codebook is at the discretion of the encoder. The encoder should take care that the indices correspond to the correct codevector entry within the transmitted order codebook table.

[0256] Cbk[class_num][k ]=sample( 8 bits );

[0257] Codebook updates are sent as run-length encoded differences between corresponding blocks in the first codebook and the second codebook. One set of context models are created for each class. A first context model measures run of zeros, while the second context addresses amplitude.

[0258] Diff_cbk[c]0[v][k]=new_cbk[c][v][k]-prev_cbk[c][v][k]

[0259] The difference codebook, diff_cbk, is calculated as the sample-wise difference between the new codebook, new_vector, and the old codebook, prev_cbk. Most diff_cbk samples will be zero, followed by small amplitudes.

[0260] Specific arithmetic coding context models are created for each class of the enhancement block indices. The first context is the original index alphabet to each class sub-table. A second context is the average of the previously transmitted above and left blocks.

[0261] The arithmetic coder is reset for each strip.

[0262] PHD Decoding

[0263] PHD decoding is a subset of the encoder operation, and is precisely modeled by the encoder as illustrated in **FIG. 5***a*. Specifically, MPEG-2 decode base layer **5025** is **4110**, predictive interpolation **5030** is **4130**, classifier **5040** is **4140**, VQ decoder **5060** is **4107**, adder **5065** is **4150**, and frame buffer store **5070** is **4170**.

[0264] Codebook Generation

[0265] Virtually any codebook design algorithm can be used to generate the enhancement codebook **5140**. The codebook could also be selected from a set of universal codebooks rather than created from some training process on the video signal to be encoded. The preferred PHD vector quantization codebook design algorithm is a hybrid of the Generalized Lloyd Algorithm (GLA), Pair-wise Nearest Neighbor (PNN), and BFOS algorithms described in [Garrido95]. The hybrid is continuously applied to each video scene. Training sequences **5130** are derived from a set of filtered HD images **5160, 5012**, rather than original HD images **5007, 5170**. Although it would be less expensive not to have the pre-processing stage **5010**, the original HD source images are not used for comparison since it may contain data patterns that are either unnecessary for the application, or unrealistic to approximate with PHD coding. The difference signal **5332, 5037** generated as the difference between the cleaned signal **5014** stored in **5013, 5160** and the interpolative-predicted signal **5032** is then fed to the codebook generator **5340**.

[0266] A potential codebook **5140** is transmitted along with each scene, where it is then parsed by the PHD decoder at the receiver side, and stored in long term memory **5160** for application throughout a scene or, in special cases, applied repeatedly in future scenes.

[0267] Syntax

[0268] The PHD syntax is structured to a hierarchy (**FIG. 7***e*) resembling traditional video coding layers known for efficient and robust parsing. A scene roughly corresponds to a typical video sequence (**FIG. 7***h*), and in addition to codebook updates, includes the energy threshold parameters **5317, 5327** used in the classification stages. Picture headers enhancement_picture( ) delineate sets of indices corresponding to the enhancement blocks for a given picture. The picture header identifies the current enhancement picture number, picture_number, and the picture payload comprises one or more strips that select which codebook codebook_number is to be applied for those blocks contained within the strip.

[0269] Referencing Multiple Codebooks

[0270] Duration of Codebook:

[0271] A codebook is created for application upon a scene which typically lasts from half a second to several seconds, such as **8210, 8220**, and **8230** depicted in **FIG. 8***c*. In extreme cases, the lengths of scenes can range from a few pictures to several minutes (thousands of pictures). Since every scene has unique image statistics and characteristics, codebooks optimized for each scene will produce better quality results for a given index rate. The overhead of sending codebooks also significantly impacts the quality-rate tradeoff. Frequent transmission of codebooks will offset the index quality gains, and potentially penalizing quality in the base bitstream (if the base stream is jointly optimized), or leave less room for future codebooks on the disc volume. Some scene changes, such as camera angle cuts with similar background (e.g. two characters talking to each other) may precipitate codebooks that largely overlap with previously

sent codebooks. The differential and dynamic codebook update mechanisms disclosed herein address these cases. Pointers to previously sent codebooks (**FIG. 8***e*) may also be more efficient for short, repeating scenes.

[0272] The PHD advantage of exploiting long-term correlations is partly illustrated in **FIG. 8***c* by the ability of a codebook (aligned to a scene) to span periods exceeding the nominal enforced "group of pictures" (GOP) dependency periods, and thus saves bits compared to a strategy where codebook are automatically sent for each GOP. Thus, for example instead of transmitting a codebook every 0.5 seconds—the period of the Intra-picture or GOP—the codebook need only be transmitted every few seconds. The random access period for the enhancement layer will thus consequentially be greater than the base layer, but as long as a base layer picture can be built with the normal short latency, a good approximation for the purposes of non-predetermined trick play can be satisfied. New codebooks are forced by the DVD authoring tools for pre-determined jumps within the DVD volume, such as angle or layer changes. Thus playback along the pre-constructed linear stream timeline will maintain constant enhanced picture quality.

[0273] In this invention, GOP is applied more widely to mean independently decodable collection of pictures, typically constructed in MPEG video stream to facilitate random access and minimize DCT drift error. "group_of_pictures( )" has a narrower meaning in the MPEG video specification than this description, but fits within the definition given here. For this invention, GOP is a generic term, and superset of the formal MPEG definition, that delineates any collection of dependently coded pictures. The duration of the GOP is typically 0.5 seconds in DVD applications, but the exact boundary of a GOP may be adjusted for scene changes or coding efficiency.

[0274] Random access to a codebook can be optimized for scene changes, buffer statistics, chapter marks, and physical models such as location of the scene data within the disc.

[0275] Nominally, multiple bitstream types such as audio, video, subpicture are time division multiplexed (TDM) within a common DVD program stream. Data for each stream type is buffered before decoding by each of the respective stream type decoders. As illustrated in **FIG. 8***d*, these buffers can allow extreme variation in the time in which coded data corresponding to one frame enters the buffer, and the time when it is later decoded and presented (e.g. to display). For purposes of buffer modeling, these stream types are deemed concurrent, although are actually serially multiplexed at the granularity of a stream packet. If a concurrent multiplex of the codebook would adversely affect other concurrent stream types (video, audio), such leaving too little bits for other concurrent streams, the encoder may send the codebook far ahead in time during a less active period of the base layer.

[0276] Multiplex Method

[0277] The majority of DVD payload packets are consumed by a single MPEG-2 System Program Stream comprising a multiplex of Packetized Elementary Streams (PES) as depicted in **FIG. 8***a*. DVD packets (**8004, 8006, 8008, 8010, 8012, 8014, 8016**, etc) are 2048 bytes long, but other non-DVD applications to which PHD are applicable may have other fixed or variable packet lengths. The flexible aspects of the of the DVD cell **8002, 8102** structure (buffering, type order and frequency) are determined by the DVD author. The example cell **8002** demonstrates the dominance of video packets owing to the larger share of the bitstream consumed by video. The actual order of packet types within the stream is arbitrary, within the limitations of buffering prescribed by the DVD standard and other standards incorporated by reference such as MPEG-2. Each concurrent data type within a DVD title is encapsulated in the multiplex as a separate PES. The program stream is an assembly of interleaved concurrent PES stream packets. The standard definition video signal (packets **8006, 8008, 8016**) is coded, as per DVD specification, with certain parameter restrictions on the MPEG-2 video tool and performance combination well known as the "Main Profile @ Main Level" (MP@ML). Other data types include Dolby AC-3 (**8008**), Sub-picture (**8014**), and navigation (**8004**) layers. Each PES stream is given unique identifier in the packet header. Room in the ID space was reserved for future stream types to be uniquely identified through the RID (Registered Stream ID) mechanism maintained by, for example, the SMPTE Registration Authority (SMPTE-RA).

[0278] PHD would appear as an additional private stream type within the multiplex (**FIG. 8***b*), with an identifying RID. Because they appear as a private stream type, PHD packets can be ignored by older DVD players without consequence to the reconstructed MP@ML base layer video. Other multiplexing schemes such as MPEG-2 Transport Stream (TS), IETF RTP, TCP/IP, UDP, can be adapted to encapsulate PHD enhancement stream appropriate for each application. MPEG-2 TS, for example, are suited for broadcast applications such as satellite, terrestrial, and digital cable television, while RTP might be chosen for streaming over the internet or a Ethernet LAN. Program Streams are required by the DVD-Video specification, whereas emerging DVD formats such as Blu-Ray have adopted MPEG-2 Transport Streams as the multiplex format.

[0279] Codebooks are a significant portion of the PHD enhancement stream. A new codebook or codebook update is optionally downloaded at the beginning of each scene. The other major portion of the enhancement stream comprise indices for coded enhancement blocks.

We claim:

1. A method of enhancing picture quality of a video signal, said method comprising the steps of:

receiving base images of pictures having a first definition from a base layer decoder;

coding the differences between said base images of pictures having a first definition and pictures having a second definition using vector quantization;

creating a database of codebooks based upon said differences between said base images of pictures having a first definition and pictures having a second definition; and

generating enhanced images based upon said base images of said pictures having a first definition and enhancement stream data.

2. The method of claim 1 further comprising a step of generating interpolated blocks based upon said base images.

3. The method of claim 1 further comprising a step of classifying said base images.

4. The method of claim 3 wherein said step of classifying said base image comprises assigning a class number and a codevector index to each region of said base image.

5. The method of claim 1 wherein said step of creating a database of codebooks comprises generating a codebook table.

6. The method of claim 5 wherein said step of generating a codebook table comprises a step of classifying image areas having common codevectors.

7. The method of claim 2 further comprising a step of generating a difference block.

8. The method of claim 7 wherein said step of generating enhanced images comprises adding interpolated blocks and said difference blocks.

9. The method of claim 1 further comprising a step of generating an enhancement stream containing enhancement data.

10. The method of claim 1 wherein said step of receiving base images of pictures having a first definition comprises receiving base images coded with a transform coder.

11. The method of claim 1 wherein said step of receiving base images of pictures having a first definition comprises receiving base images coded with MPEG on a DVD.

12. The method of claim 1 wherein said step of analyzing the differences between said base images of pictures having a first definition and pictures having a second definition using vector quantization comprises a step of using a Generalized Lloyd Algorithm.

13. The method of claim 1 wherein said step of analyzing the differences between said base images of pictures having a first definition and pictures having a second definition using vector quantization comprises a step of using a Pair-wise Nearest Neighbor Algorithm.

14. The method of claim 1 wherein said step of analyzing the differences between said base images of pictures having a first definition and pictures having a second definition using vector quantization comprises a step of using a BFOS algorithm.

15. The method of claim 1 wherein said step of analyzing the differences between said base images of pictures having a first definition and pictures having a second definition using vector quantization comprises a step of using a combination of a Generalized Lloyd Algorithm, a Pair-wise Nearest Neighbor Algorithm, and BFOS algorithm continuously applied to each screen of said video signal.

16. The method of claim 1 wherein said step of receiving base images of pictures having a first definition comprises receiving standard definition picture having a resolution from a group consisting of:

720×480;

704×480

704×576; and

720×576.

17. The method of claim 18 wherein said step of coding the differences between said base images of pictures having a first definition and pictures having a second definition comprises coding the differences between said base images of pictures having a first definition and pictures having a resolution from a group consisting of:

1920×1080;

1440×960

1440×1152; and

1920×1152.

18. The method of claim 1 wherein said step of generating enhanced images based upon said base images of standard definition pictures and enhancement stream data comprises a step of generating enhanced images based upon said base images of standard definition pictures, codebook data and codevector indexes.

19. A method of enhancing picture quality of a video signal, said method comprising the steps of:

analyzing the differences between said image of standard definition pictures and high definition pictures;

creating a database of codebooks based upon said differences between said images of standard definition pictures and high definition pictures;

receiving base images of standard definition pictures from a base layer decoder;

generating an interpolated block based upon said base images of standard definition pictures;

generating a difference block based upon said codebook; and

generating enhanced images based upon said standard definition images by adding said interpolated block and said difference block.

20. The method of claim 19 further comprising a step of classifying said base images.

21. The method of claim 20 wherein said step of classifying said base image comprises assigning a class number to each region of said base image.

22. The method of claim 19 wherein said step of creating a database of codebooks comprises generating a codebook table.

23. The method of claim 22 wherein said step of generating a codebook table comprises a step of classifying images having common codevectors.

24. The method of claim 22 further comprising a step of encoding a codevector index.

25. A circuit for enhancing picture quality of a video signal, said circuit comprising:

a base layer decoder generating a base image of a standard definition picture;

an interpolator coupled to said base layer decoder and generating an interpolated block;

a classifier coupled to said base layer decoder and generating a class number; and

a summing circuit coupled to said interpolator and said classifier, said summing circuit adding said interpolated block and a difference block.

26. The circuit of claim 25 wherein said interpolator comprises a temporal predictive interpolator.

27. The circuit of claim 25 wherein said interpolator comprises a circuit for providing motion compensation.

28. The circuit of claim 25 further comprising a second summing circuit coupled to said classifier and an index from an enhance stream decoder.

29. The circuit of claim 25 further comprising a codebook table coupled to said second summary circuit.

**30**. The circuit of claim 29 wherein said codebook tables comprise classes of codevectors.

**31**. The circuit of claim 30 wherein said classes are based upon properties measured on base images and previously enhanced images in the decoder.

**32**. The circuit of claim 25 further comprising an enhanced picture based upon said base image of a standard definition picture.

**33**. A circuit for enhancing picture quality of a video signal, said circuit comprising:

base layer decoder means generating a base image of a standard definition picture;

temporal predictive interpolator means coupled to said base layer decoder means and generating an interpolated block;

classifier means coupled to said base layer decoder means and generating a class number; and

summing circuit means coupled to said temporal predictive interpolator means and classifier means, said summing circuit means adding said interpolated block and a difference block.

* * * * *

US 20050114909A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2005/0114909 A1

Mercier

(43) Pub. Date: May 26, 2005

(54) HIGH DEFINITION MEDIA STORAGE STRUCTURE AND PLAYBACK MECHANISM

(76) Inventor: Guillaume Mercier, McLean, VA (US)

Correspondence Address:
DAVID G GROSSMAN
1408 BAYSHIRE LANE
HERNDON, VA 20170 (US)

(21) Appl. No.: 11/017,927

(22) Filed: Dec. 22, 2004

Related U.S. Application Data

(63) Continuation of application No. 09/540,557, filed on Mar. 31, 2000, now Pat. No. 6,865,747.

(60) Provisional application No. 60/127,394, filed on Apr. 1, 1999.

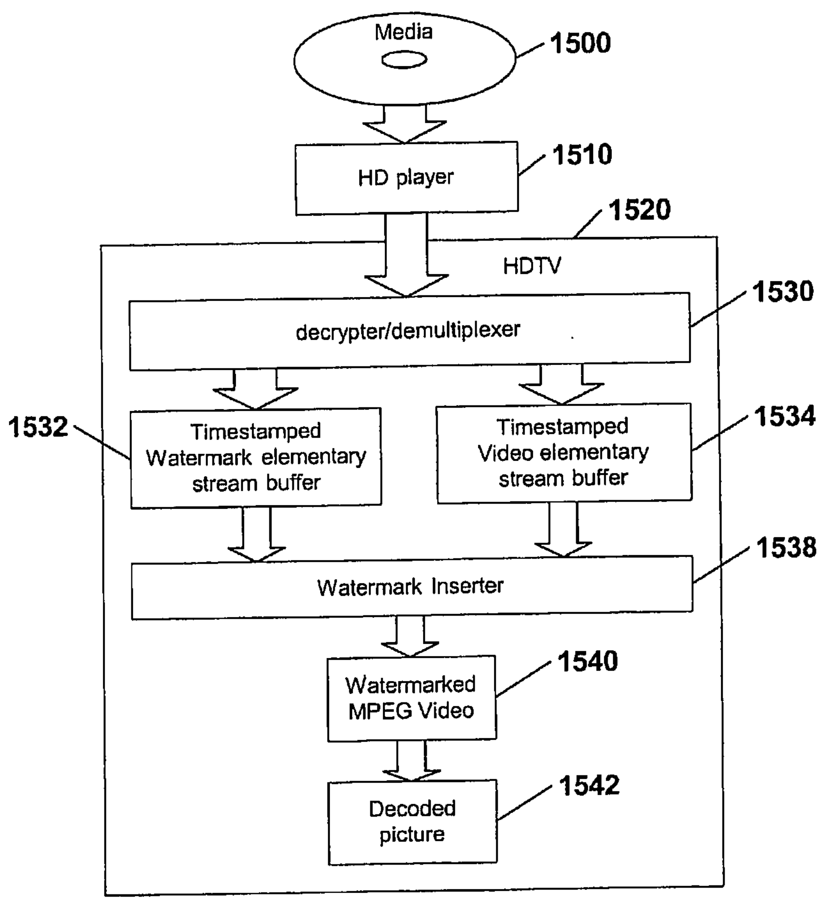
Publication Classification

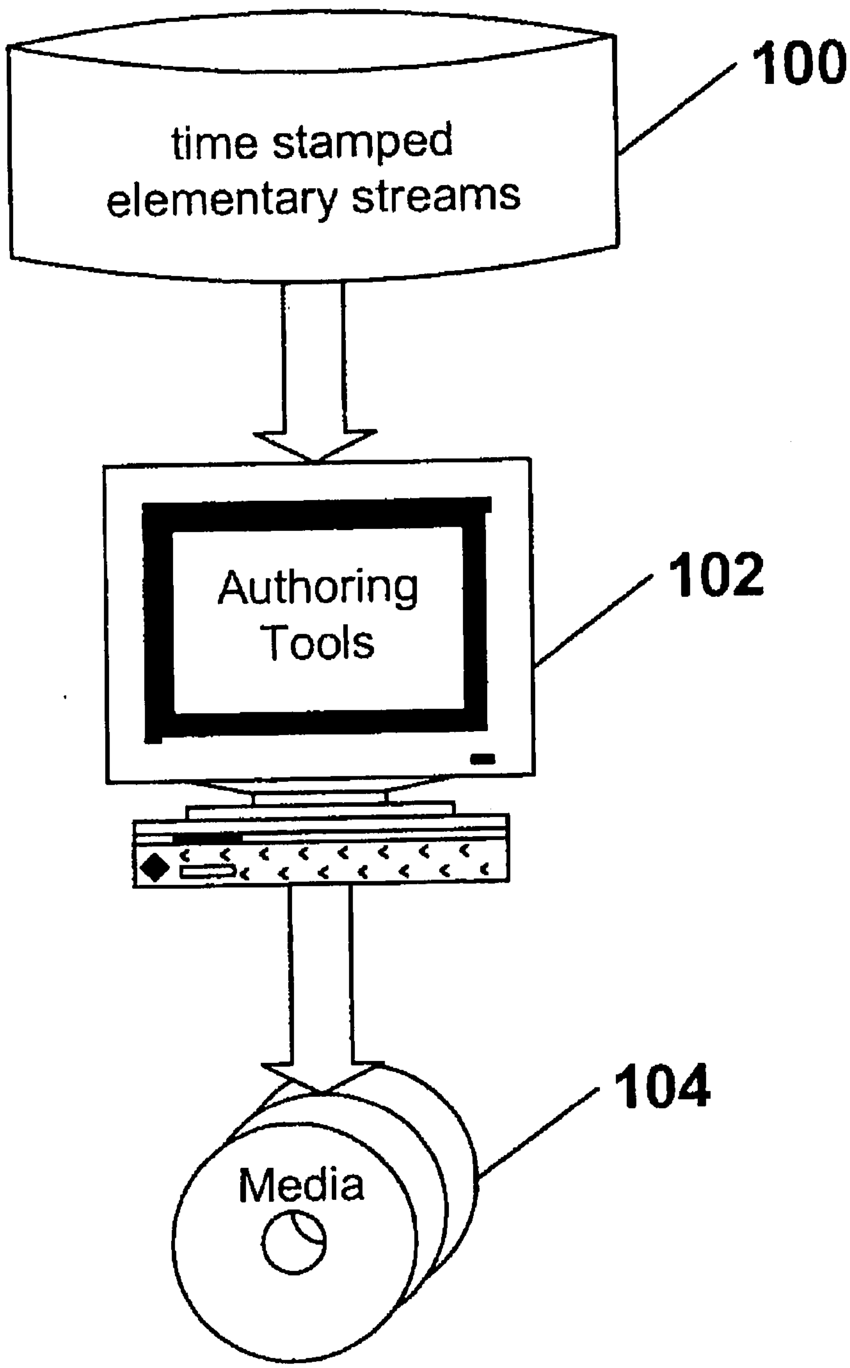
(51) Int. Cl.[7] ..... H04N 7/16

(52) U.S. Cl. ..... 725/141

(57) ABSTRACT

An apparatus and method for storing and playing high definition content is disclosed. This invention provides a mechanism for storing and playing back high definition content on a medium such as DVD optical disc. One aspect of the invention is that elementary streams may be multiplexed and processed in a high definition media player instead of at authoring time. Another aspect of the invention is that it provides for extended real-time features such as inserting watermarks into the content stream, decrypting selected sections of the content stream, and performing trick playback display modes.





time stamped elementary streams
100
Authoring Tools
102
Media
104


FIG. 1

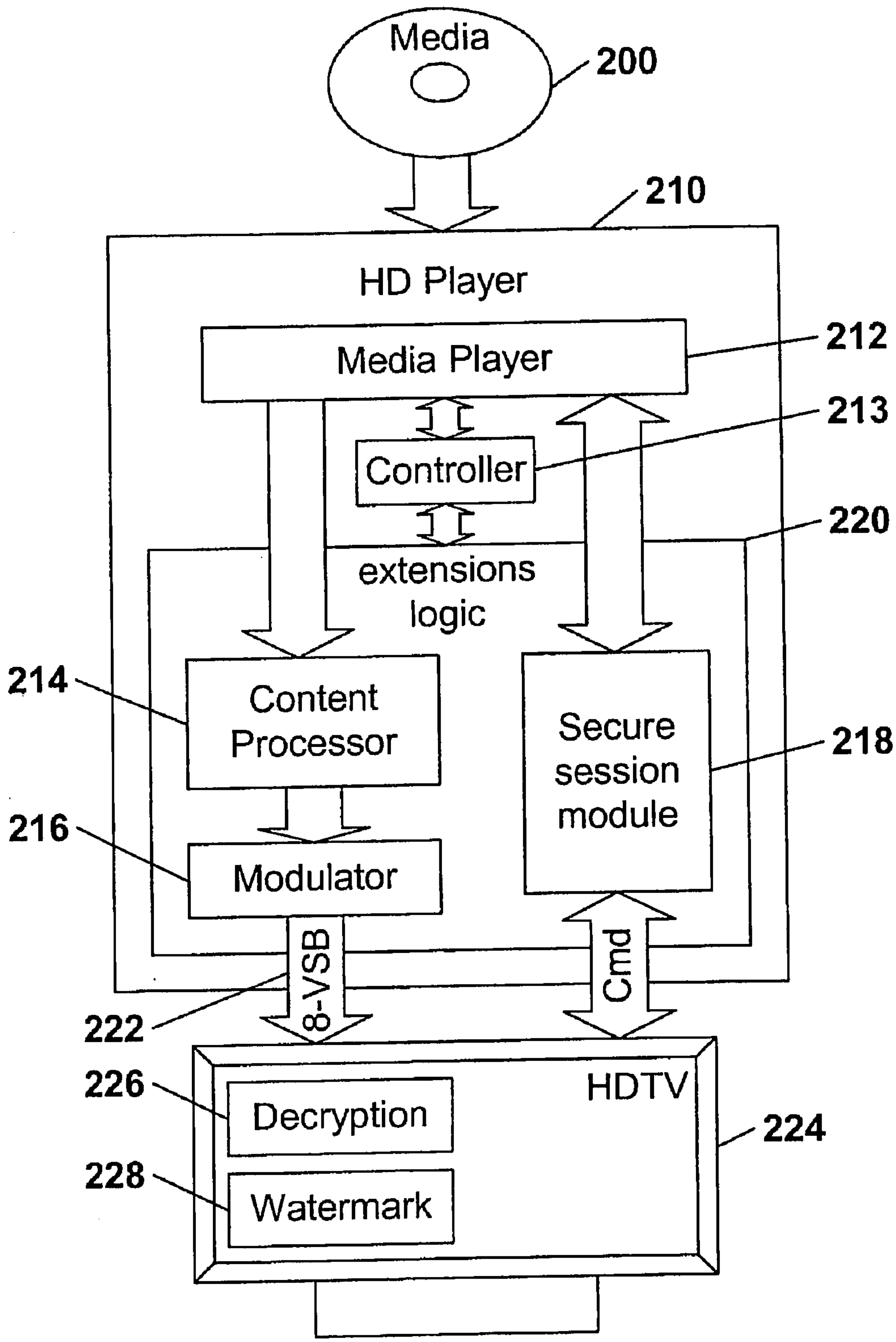

Media
200
210
HD Player
Media Player
212
213
Controller
220
extensions logic
214
Content Processor
Secure session module
218
216
Modulator
8-VSB
Cmd
222
226
HDTV
Decryption
224
228
Watermark

FIG. 2

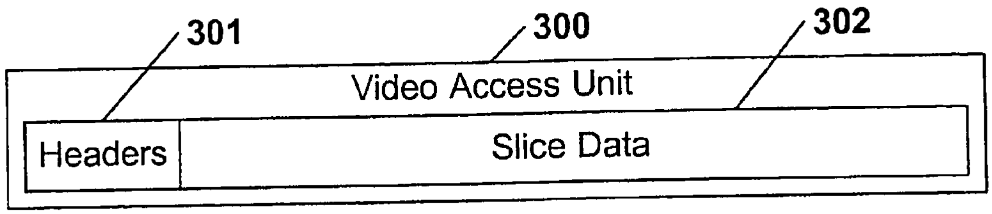



FIG. 3A

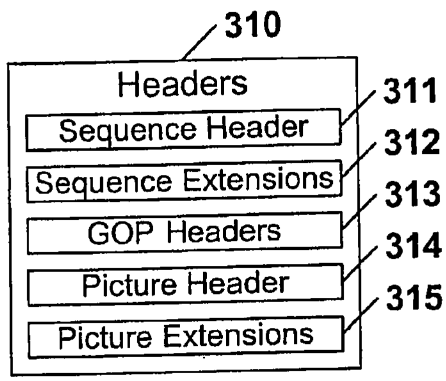



FIG. 3B

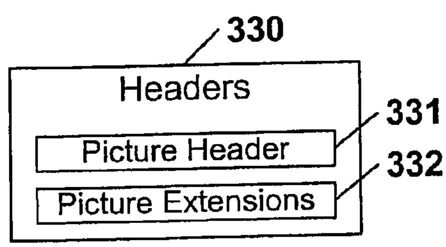



FIG. 3C

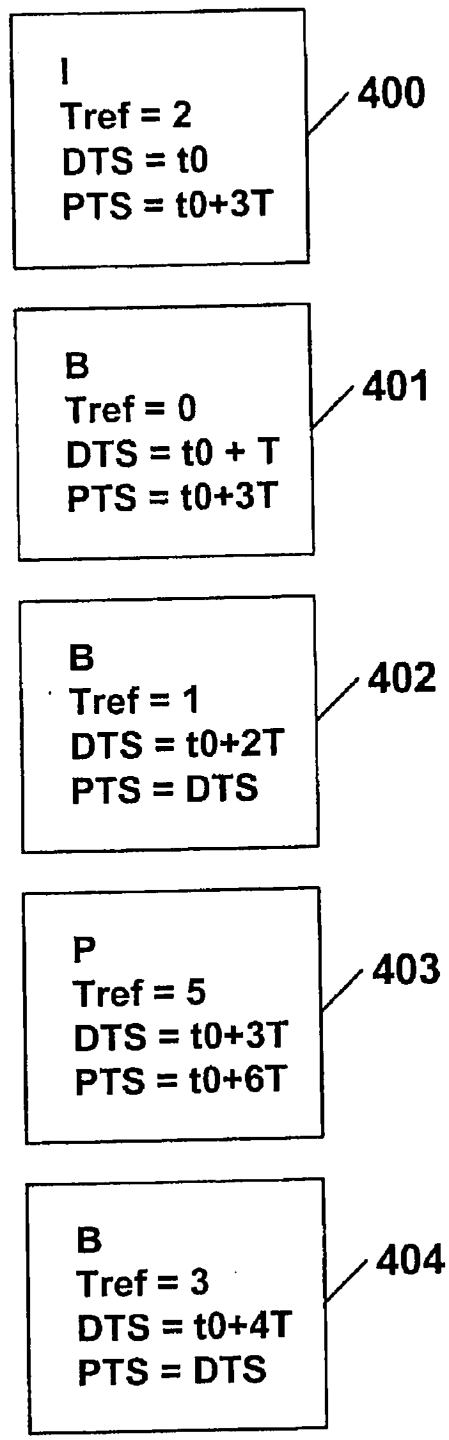

I
Tref = 2
DTS = t0
PTS = t0+3T
400
B
Tref = 0
DTS = t0 + T
PTS = t0+3T
401
B
Tref = 1
DTS = t0+2T
PTS = DTS
402
P
Tref = 5
DTS = t0+3T
PTS = t0+6T
403
B
Tref = 3
DTS = t0+4T
PTS = DTS
404

FIG. 4

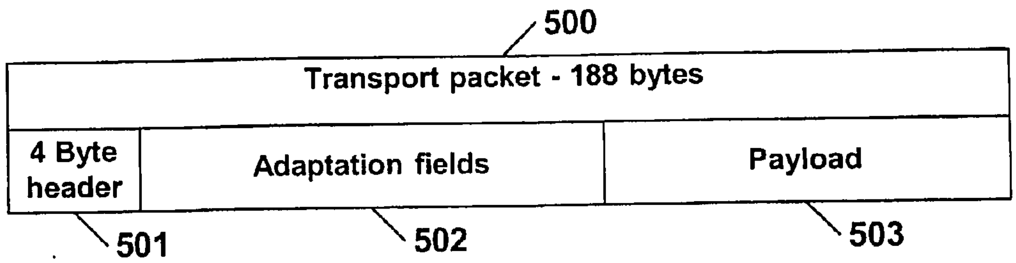



FIG. 5

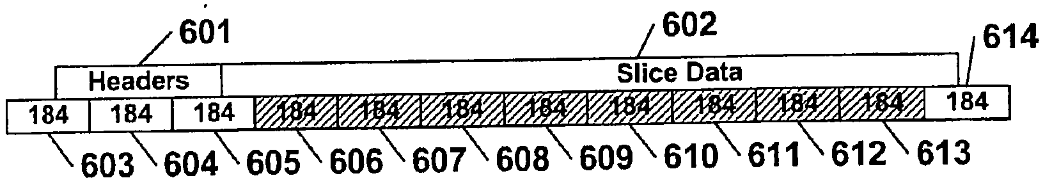



FIG. 6

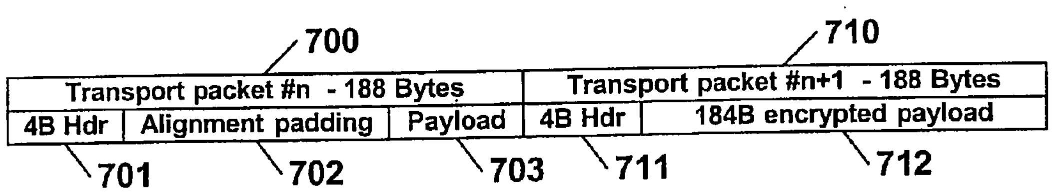



FIG. 7

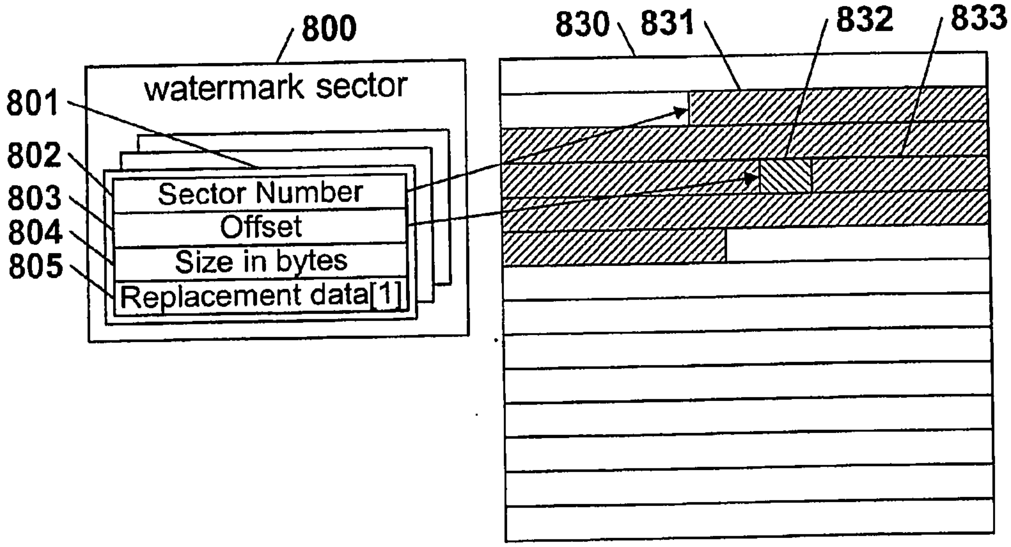

800
watermark sector
801
802
803
804
805
Sector Number
Offset
Size in bytes
Replacement data[1]
830
831
832
833

FIG. 8

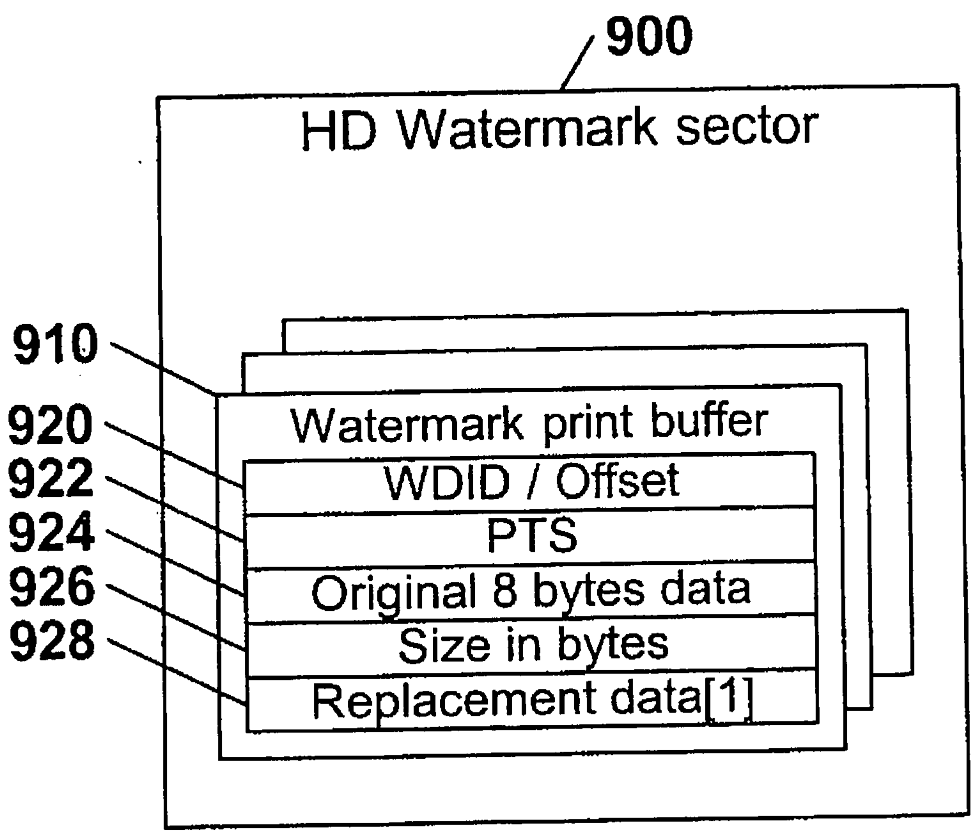

900
HD Watermark sector
910
920
922
924
926
928
Watermark print buffer
WDID / Offset
PTS
Original 8 bytes data
Size in bytes
Replacement data[1]

FIG. 9

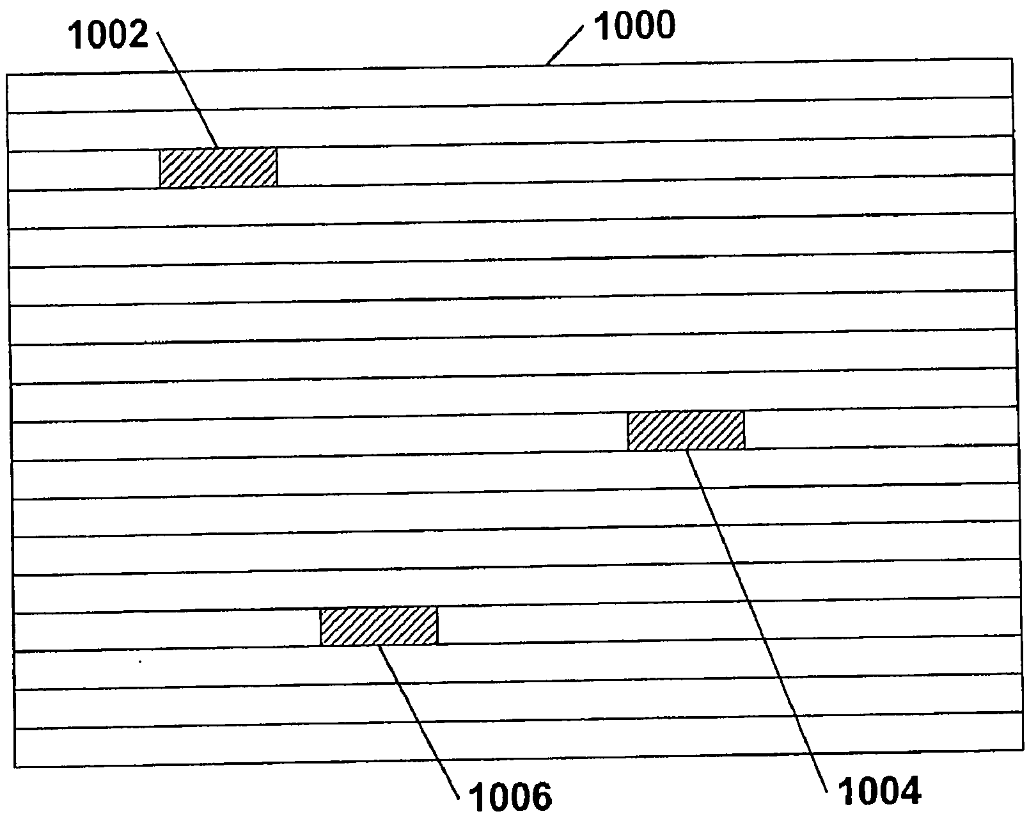



FIG. 10A

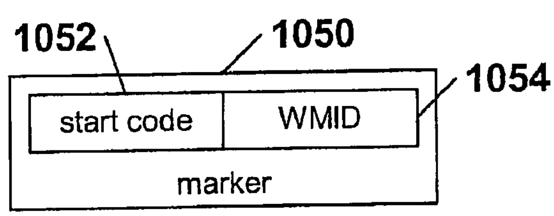



FIG. 10B

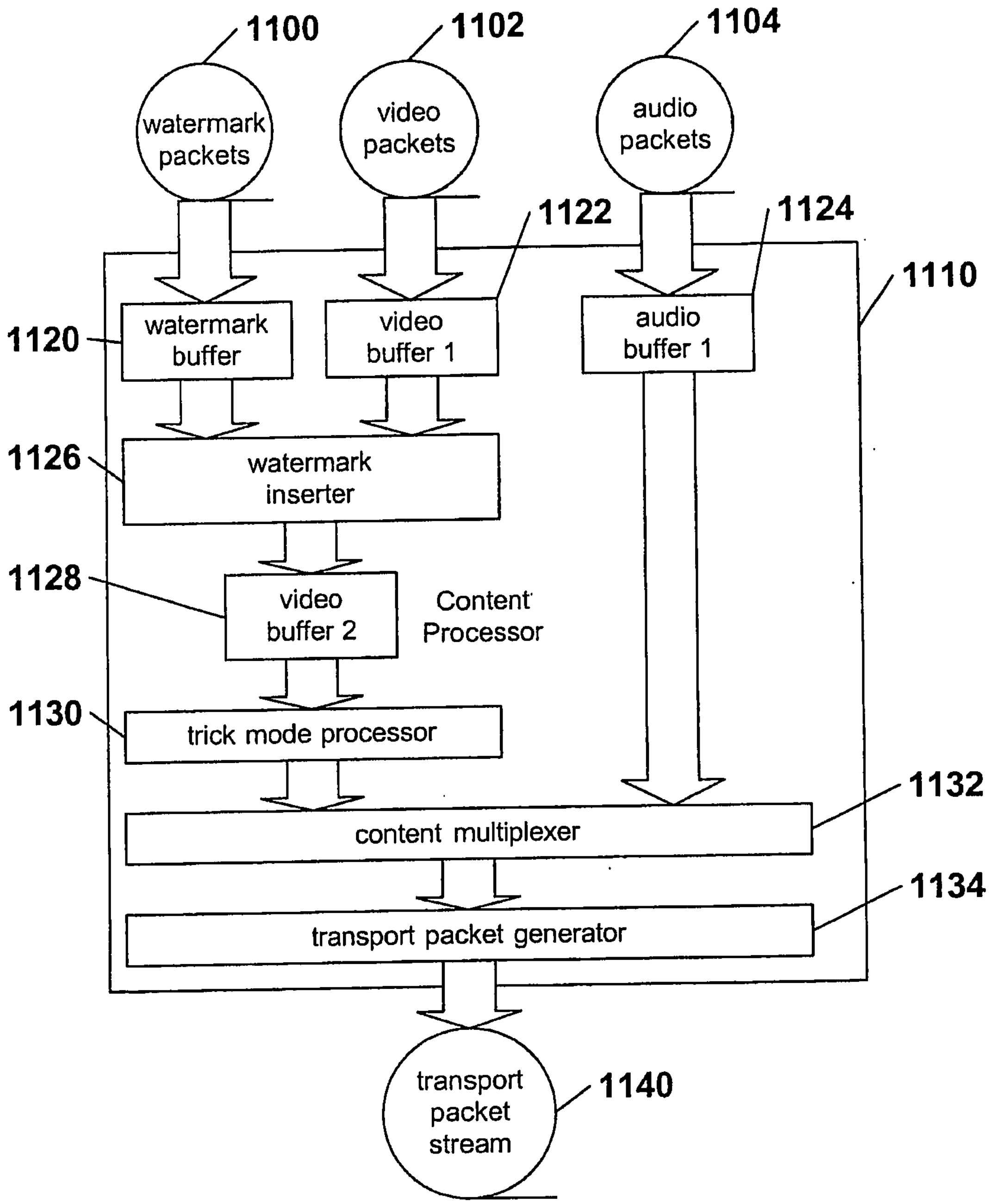

1100
watermark packets
1102
video packets
1104
audio packets
1122
1124
1110
1120
watermark buffer
video buffer 1
audio buffer 1
1126
watermark inserter
1128
video buffer 2
Content Processor
1130
trick mode processor
1132
content multiplexer
1134
transport packet generator
transport packet stream
1140

FIG. 11

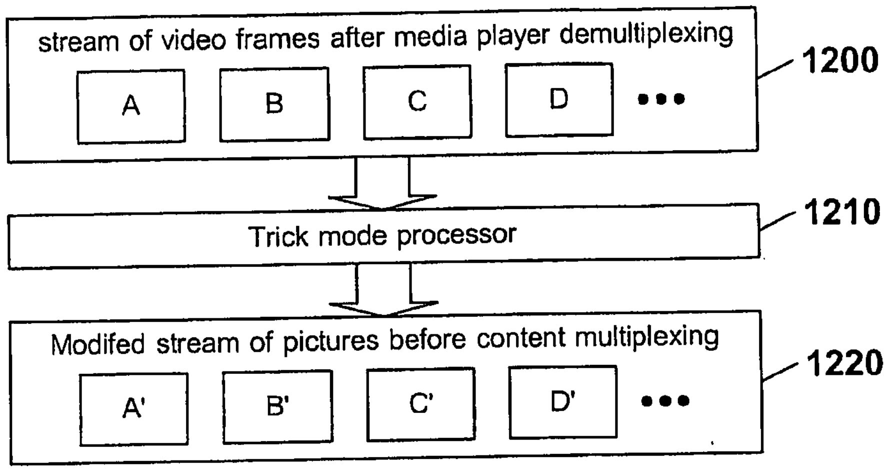

stream of video frames after media player demultiplexing
A
B
C
D
•••
1200
Trick mode processor
1210
Modifed stream of pictures before content multiplexing
A'
B'
C'
D'
•••
1220

FIG. 12

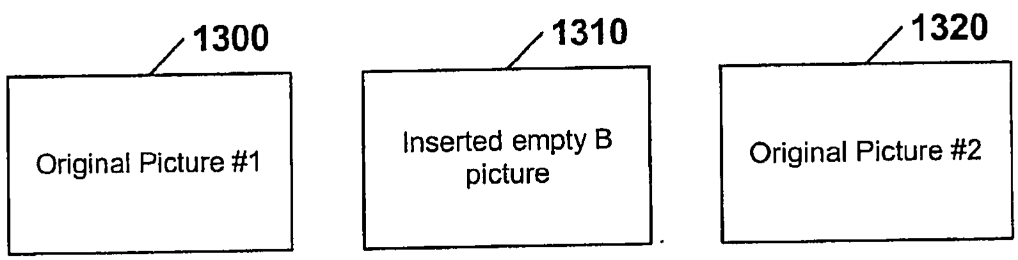

1300
Original Picture #1
1310
Inserted empty B picture
1320
Original Picture #2

FIG. 13

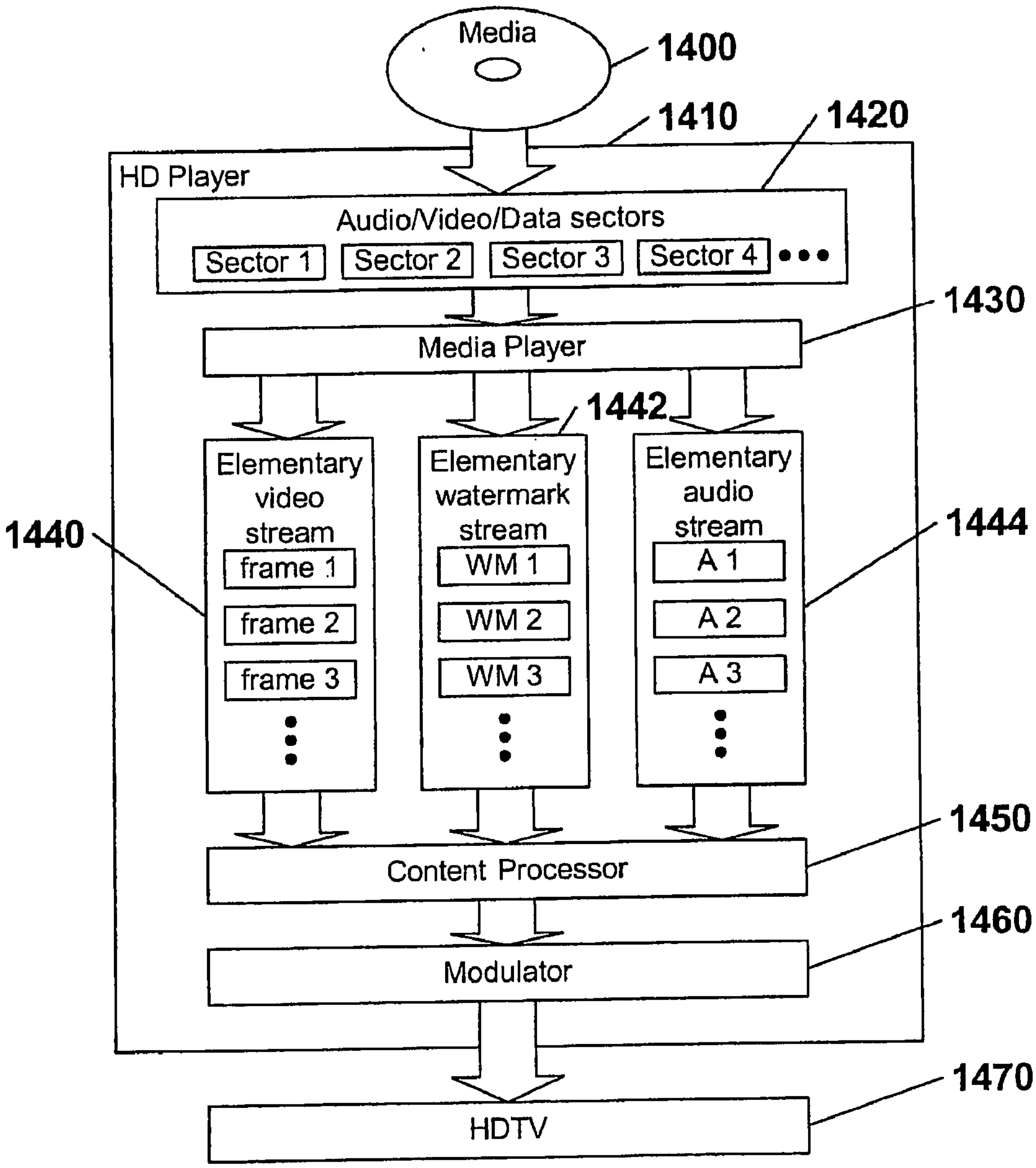

Media
1400
1410
1420
HD Player
Audio/Video/Data sectors
Sector 1
Sector 2
Sector 3
Sector 4
1430
Media Player
1442
Elementary video stream
frame 1
frame 2
frame 3
Elementary watermark stream
WM 1
WM 2
WM 3
Elementary audio stream
A 1
A 2
A 3
1440
1444
1450
Content Processor
1460
Modulator
1470
HDTV

FIG. 14

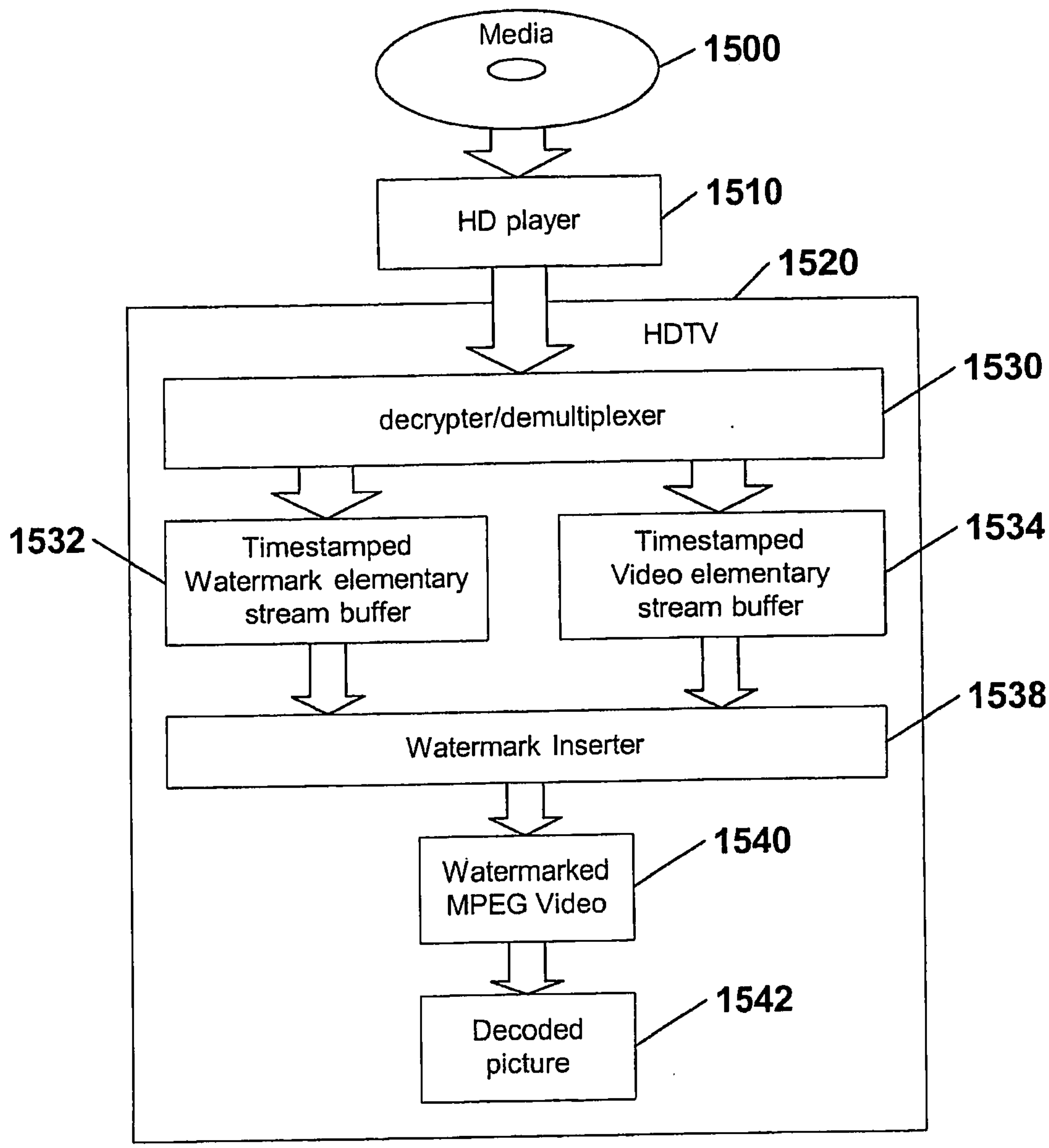

Media
1500
HD player
1510
1520
HDTV
decrypter/demultiplexer
1530
1532
Timestamped Watermark elementary stream buffer
Timestamped Video elementary stream buffer
1534
Watermark Inserter
1538
Watermarked MPEG Video
1540
Decoded picture
1542

FIG. 15

# HIGH DEFINITION MEDIA STORAGE STRUCTURE AND PLAYBACK MECHANISM

[0001] The present application claims priority on co-pending commonly assigned provisional patent application Ser. No. 60/127,394, to Mercier, filed on Apr. 1, 1999, entitled "High Definition Digital Video Disc Format", the contents of which are incorporated by reference herein.

## FIELD OF THE INVENTION

[0002] The present invention relates to high definition media storage structures and playback mechanisms.

## BACKGROUND OF THE INVENTION

[0003] Mechanisms for storage and processing of digital content on various media have been defined for various digital content playback systems. Recently, the resolution of digital content has increased. This content is now referred to as high definition digital content (HDDC). Current storage structures and playback mechanisms were not designed specifically for HDDC. There is a need for new storage structures and playback mechanisms for HDDC that introduce as little impact on current storage structures and play back mechanisms as possible. These new storage structures and playback mechanisms will preferably support methods to prevent unauthorized access to the HDDC and to track any unauthorized access to HDDC. It is also desired that these new structures and playback mechanisms will support trick playback modes. The present invention broadly relates to and provides a solution to these problems.

[0004] While the description which follows may sometimes be described in the context of audio/video/data as an example of content, the invention is not so limited and may equally apply to any type of information or content data, including without limitation audio and/or video data or other type of data or executables.

[0005] The invention is described in terms of the current best mode. This best mode is described as extensions of the DVD Specifications for Read-Only Disc (described in "DVD Specifications for Read-Only Disc", Version 1.1, December 1997 by Toshiba Corporation) to support high resolution, encrypted and actively watermarked content. Media conforming to these extensions are referred to in this document as HD-DVD. Playback mechanisms which present the HDDC content to an ATSC/HDTV compatible receiver are also disclosed. These mechanism allow graphics, trick modes, and watermarking to be extended to HDTV. One skilled in the art can see that although the present invention is described in terms of HD-DVD, the invention may be practiced on any digital storage media including hard disks, magnetic tape, and other optical discs.

[0006] The present application is directed to the same general technology as co-pending commonly assigned patent application Ser. No. PCT/US00/00079, entitled "Content Packet Distribution" naming Schumann et al. as inventors (the contents of which are incorporated by reference herein). This application is directed more to specific storage structures and playback mechanisms including watermark insertion, trick modes, and ATSC stream generation.

[0007] In some commercial applications, where the content includes, for example, valuable audio or video content, unauthorized access by those who obtain the content may tend to reduce the profit margin of the content provider(s), who typically provide the content, e.g. to various listener and/or viewers, for a fee. In particular, with the advent of high definition video, this problem is even more serious because the digital data is of sufficient resolution to be shown on a full size theater screen. This opens up a whole new area for content pirates to market their stolen property. If the unauthorized accesser is a content pirate, he or she may pose a serious threat to a content provider by inducing others to pirate the content as well. More particularly, the pirate may generally sell pirated access to the content at a lower cost than the legitimate content provider because the pirate obtains access to the content by using the legitimate provider's infrastructure and therefore does not have to invest resources to produce and disseminate the content. This becomes even a greater concern where the pirate may copy and mass produce a relatively inexpensive component which allows a large number of users to obtain access to the content without authorization by the legitimate content provider. As a result, content providers have resorted to increasingly expensive and complex schemes to prevent unauthorized access to their information and content, i.e. to prevent pirating.

[0008] What is needed is a system and method for protecting valuable content; a method and system which is robust, which may be tailored to the needs of a particular content provider, and which overcomes the above noted deficiencies.

## SUMMARY AND ADVANTAGES OF THE INVENTION

[0009] One advantage of the invention is that it allows a disc to be authored where the disc may be played by both conventional media players and high definition media players.

[0010] Another advantage of this invention is that elementary streams may be multiplexed and processed in the high definition media player instead of at authoring time.

[0011] Yet a further advantage of this invention is that it provides for extended real-time features such as inserting watermarks into the content stream, decrypting selected sections of the content stream, and performing trick modes.

[0012] To achieve the foregoing and other advantages, in accordance with all of the invention as embodied and broadly described herein, an apparatus for playing high definition content comprising a media player for receiving the high definition content from a media source. The high definition content is contained in data packets and the data packets are contained in sectors. A content processor processes the high definition content into transport packets and a transport packet modulator modulates the transport packets. A controller manages the operations of the apparatus.

[0013] In yet a further aspect of the invention, the apparatus for playing high definition content further includes a watermark buffer for receiving watermark data; a video buffer for receiving video data; an audio buffer for receiving audio data; a watermark inserter for inserting watermarks into the video data, determined by the video data and watermark data; a content multiplexer; and a transport packet generator.

[0014] In yet a further aspect of the invention, preselected bocks of the data packets are encrypted.

[0015] In yet a further aspect of the invention, the apparatus further includes a trick mode processor that can: create a slow motion effect by inserting empty predictive frames into a video elementary stream between picture frames; create a pause effect by iteratively inserting into the video elementary stream a sequence comprising an Intra-coded picture frame; and a multitude of predictive frames; create a fast forward playback effect by inserting forwardly sequenced Intra-coded picture frames interspersed with empty predictive frames into the transport packet stream; and create a rewind playback effect by inserting reverse sequenced Intra-coded picture frames interspersed with empty predictive frames into the transport packet stream.

[0016] In a further aspect of the invention, a method for playing high definition content comprising: receiving the high definition content from a media source, the high definition content contained in data packets and the data packets contained in sectors; processing the high definition content into transport packets; modulating the transport packets; and outputting the modulated transport packets.

[0017] Additional objects, advantages and novel features of the invention will be set forth in part in the description which follows, and in part will become apparent to those skilled in the art upon examination of the following or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The accompanying drawing, which are incorporated in and form a part of the specification, illustrate an embodiment of the present invention and, together with the description, serves to explain the principles of the invention.

[0019] **FIG. 1** is a block diagram of a high definition content authoring system.

[0020] **FIG. 2** is a block diagram of an embodiment of an aspect of the present invention used to playback high definition content.

[0021] **FIG. 3A** is a block diagram showing an example of a video data packing format.

[0022] **FIG. 3B** is a block diagram showing an example of a video data packing format header.

[0023] **FIG. 3C** is a block diagram showing another example of a video data packing format header.

[0024] **FIG. 4** is a diagram depicting timestamp calculations from a video bit stream.

[0025] **FIG. 5** is a block diagram of an ATSC transport packet.

[0026] **FIG. 6** is a block diagram showing how video access unit data may be encrypted as per an embodiment of the invention.

[0027] **FIG. 7** is a block diagram showing alignment of encryped data in the transport payload as per an embodiment of the invention.

[0028] **FIG. 8** is a block diagram of watermark sectors as performed by some current non-HD systems.

[0029] **FIG. 9** is a block diagram of HD watermark sectors as performed by an exemplary aspect of the present invention.

[0030] **FIG. 10A** is a block diagram of an exemplary aspect of the present invention depicting watermark markers in a frame of video data.

[0031] **FIG. 10B** is a block diagram of an exemplary aspect of the present invention depicting a watermark marker structure.

[0032] **FIG. 11** is a block diagram of an exemplary aspect of the present invention depicting a content processor.

[0033] **FIG. 12** is a block diagram showing trick mode processing.

[0034] **FIG. 13** is a block diagram showing how the slow motion playback trick mode can be obtained by inserting empty B pictures.

[0035] **FIG. 14** is a block diagram showing data flow through an embodiment of the present invention.

[0036] **FIG. 15** is a block diagram showing an embodiment of the present invention wherein watermarks are inserted into the content in the HDTV.

## DETAILED DESCRIPTION OF THE INVENTION

[0037] The present invention provides for storing high definition content on a DVD or other storage media by extending the current specification of DVD read-only disc. The global disc layout may remain identical, preserving software investments for DVD authoring tools & player firmware, but higher video resolution and bit rates are allowed. HD-DVD players may not need any MPEG-2 video or AC-3 audio decoders, but may use instead a real-time content processor **214** and a modulator **216**. This invention also provides for various "trick modes", including fast forward, reverse, slow motion, and pause. Also provided for are mechanisms that may allow the content to be encrypted and watermarked.

[0038] Encryption may be done on video, audio or other elementary streams during authoring, and may be based on blocks of consecutive bytes. Alignment methods ensures the mapping of encrypted blocks to the payload of a transport packet, and some rules define the conditions under which a block may or may not be encrypted, and where an encrypted block has to start. The transmission of watermarks in encrypted format to the TV receiver follows a buffering method and individual watermarks may be grouped in time stamped access units. Trick modes are also possible by slightly altering the content of video access unit headers and by inserting or suppressing MPEG-2 video frames. Finally, backward compatibility of, the new system is possible if the audio and video formats of the classic DVD are supported by the ATSC standard (AC-3 audio, MPEG-2 video). MPEG graphics may also be supported.

[0039] Reference will now be made in detail to the presently preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings.

[0040] A block diagram of a high definition content authoring system is shown in **FIG. 1**. Authoring systems are used to create the final image of the digital content in a

format compatible with the intended display system. Time stamped elementary input streams **100**, including audio and video input streams formatted as MPEG-2 video and AC3 audio respectively, may be multiplexed together by the authoring tool **102** as they would be with a classic DVD, including data for multiple angles and parental levels. The authoring tool **102** is in charge of creating system files for Video Manager and Video Title Sets, one or more MPEG-2 program stream including navigation packs (Video Objects), and storing them as a disc or set of discs **104**. In the presently illustrated embodiment, the discs **104** may be UDF formatted.

[0041] **FIG. 2** is a block diagram of an embodiment of an aspect of the present invention used to playback high definition content. The disc(s) **200** containing authored content may be inserted in a HD-DVD enabled player **210**. The player has a media player **212** that process the DVD specifications and extensions logic **220** that process the HD-DVD extensions. Together, the media player logic **212** and extensions logic **220** interpret the contents of the disc(s) **200** to create a combination of graphic menus and audio/video streams.

[0042] An embodiment of the present invention may include a secure session module **218**, so that secured communication may be established with the receiver before content playback may be authorized.

[0043] To send data to the receiver, the program streams stored on the disc **200**, or generated by an MPEG graphic engine are converted to an ATSC/HDTV transport stream by the content processor **214**. The stream is then modulated by a modulator **216** and sent to an HDTV receiver **224**. The present embodiment used 8-VSB modulation, however, any type of modulation capable of transporting the digital content may be used.

[0044] The receiver **224** demodulates the signal and reconstructs the transport stream packets, which are sent to their intended destinations which may include a decryption encoder **226**, audio and video decoders, or watermark logic **228**.

[0045] DVD Read Only vs. HD-DVD Specifications.

[0046] The HD-DVD extensions may closely follow the DVD Specifications for Read-Only Disc, may enlarge the range of video formats allowed and may allow higher data rates on the disc. In the embodiment of the present invention, wherein compatiblity with the ATSC/HDTV standard is desired, only those video and audio formats defined in the ATSC/HDTV standard may be used when the HD-DVD contains high definition material intended for HDTV display. The term 'HD mode' refers to when the HD-DVD player plays a disc with features not found in the DVD specifications for Read-Only disc.

[0047] A detailed description of the new parameter boundaries and constraints may be defined, in particular in the interleaved units minimum jump sizes that a HD-DVD player has to meet for multi-angle blocks. Areas that that may be extended for HD-DVD includes: parts of the video objects such as the contents of VOB, the player reference model, the presentation video, audio and sub-picture unit data; restrictions for seamless play; restrictions of SP_DC-CMDs; relation between Information in disc and player; display mode; position and allowed line number's range of video and sub-picture; and karaoke mode in MPEG-2 audio.

[0048] In HD mode, the MPEG-2 video and audio configuration (i.e. bit rate, profile, level, AC3 for audio) may need to meet the ATSC/HDTV standard requirements. Audio and video streams stored as an MPEG-2 program stream on the disc may be converted to an ATSC/HDTV compliant transport stream by the Content processor **214**.

[0049] Audio & Video Demultiplexing

[0050] Extracting audio and video access units may be performed by extracting the payload of each corresponding packet. A start code search may be done to find each access unit's boundaries, unless pointers are added on the disc as private data. As shown in **FIG. 3A**, each MPEG video access unit (VAU) **300** may include a header **301** and slice data **302**, where headers may comprise one of the two descriptions as illustrated in **FIGS. 3B and 3C**.

[0051] **FIG. 3B** is a block diagram showing an example of a video data packing format header **310** which may include a sequence header **311**, sequence extensions **312**, GOP (group of picture) headers **313**, picture headers **314** and picture extensions **315**.

[0052] **FIG. 3C** is a block diagram showing another example of a video data packing format header **330** which may include a picture header **331** and a picture extensions **332**.

[0053] An AC3 audio frame may start with a sync word OxB77 and be encoded at constant bit rate, which makes frame extraction quicker.

[0054] Timing Constraints

[0055] Timestamps include several components including a Tref, a Decoding Time Stamp (DTS), and a Presentation Time Stamp (PTS). The DTS represents the time to decompress the frame. The PTS represents the the time to present the frame. Tref is a temporal reference number. Obtaining a time stamp for each MPEG video access unit (PTS and DTS) can be done during VAU extraction, by using the PTS and DTS fields found in PES packet headers of DVD sectors. Only I frames are required to have PTS and DTS. Time stamps for other pictures can be computed, but at the expense of extra memory to store data until the next reference frame (the DTS of a reference frame should be equal to the PTS of the previous reference frame). Time stamps may also be inserted at DVD authoring time to reduce memory requirements in the player.

[0056] An example of DTS and PTS computation from a video bit stream, when only the first PTS and DTS are known is illustrated in **FIG. 4**. Timestamps **400, 401, 402, 403** and **404** represent a sequence of timestamps. Timestamp **400** corresponds to an I frame. Timestamp **403** corresponds to a P frame which is derived from an I frame. Timestamps **401, 402** and **404** correspond to B frames which may be derived by either P frames, I frames, or both.

[0057] AC3 audio streams may be encoded at constant bit rate, in which case the PTS can be computed by linear extrapolation.

[0058] Multiplexing Streams

[0059] A novel aspect of the present invention is that the transport multiplexing module may run independently in the player. Previous art usually performs this task in the authoring tool. In the presently illustrated example, there are three inputs: a video stream, an audio stream, and private data streams, each access unit being sent with a corresponding set of time stamps. In movie play mode, the streams are multiplexed according the MPEG-2 specifications, the private data stream following a buffer model similar to the audio stream. The timestamped private stream may be used to assist in watermarking the content at runtime using a corresponding watermark access unit.

[0060] In MPEG graphics mode, the background is sent as an I frame and picture elements are added using P frames. Techniques similar to trick mode play may be used to build a valid, time stamped, MPEG-2 elementary stream that may be sent to the transport multiplexing module.

[0061] Any DVD authored according to the DVD Specifications for Read Only Disc with video and audio format supported by the ATSC (MPEG-2 video and AC3 audio) may offer a valid input for the transport multiplexer **214** and may be sent to an ATSC compliant HDTV through an 8-VSB interface **222**.

[0062] Storage, Format and Procedures to Handle Encrypted Content.

[0063] Another novel feature of the present invention is that it allows content to be independently encrypted on a block by block basis. The DVD storage format is based on 2048 bytes per logical sector. This format is only used for storage, and the transmission of data may be done with ATSC transport packets of 188 bytes:

[0064] **FIG. 5** is a block diagram of a 188 byte ATSC transport packet **500**. The Transport packet **500** may include a 4 byte header **501**, an adaption field **502**, and a payload **503**. When a packet has no adaptation field **502**, the payload **503** may have a size of 184 bytes. When an adaptation field **502** is present, to carry a PCR or padding bytes for example, the number of bytes of the payload **503** may be reduced accordingly.

[0065] **FIG. 6** is a block diagram showing how video access unit data may be encrypted as per an embodiment of the invention. To allow a real-time conversion from a time stamped MPEG-2 video stream and a time stamped AC3 audio stream to a valid transport stream, some fields in the headers **601** may have to be read and/or modified. For this reason, they may not be encrypted. In a video access unit the encryption may start on the first 184 byte block **606** completely contained in the slice data area **602**, continue through blocks **607**, **608**, **609**, **610**, **611**, **612** and stop on the last 184 byte block **613** completely contained in the slice data area. Not all of these blocks have to be encrypted, but no other blocks may be encrypted in the video stream. Therefore, blocks **603**, **604**, **605** and **614** are not encrypted.

[0066] Audio streams can be encrypted in a less restrictive manner, since the size of an access unit can be predicted. For example, only 1 out of 10 audio access units can be left unencrypted.

[0067] A major problem with encryption of elementary streams is to avoid any misalignment between elementary stream encrypted data and transport streams packet decryption. **FIG. 7** is a block diagram showing alignment of encrypted data in the transport payload as per an embodiment of the invention. Transport packet **700** contains a 4-byte header **701**, alignment padding bytes **702**, and payload bytes **703**. The second transport packet **710**, includes a 4 byte header **711** and **184** bytes of encrypted payload data **712**. A solution to this problem, if it occurs, is to insert padding bytes **702** in the last packet **700** preceding a group of encrypted packet **710** to ensure the correct alignment of the 184 bytes of the transport packet payload **712** (a transport packet cannot contain both encrypted and unencrypted data). If the data is encrypted as previously described, then only one padding operation is required, in the last packet preceding the blocks of encrypted slice data. When an adaptation field must be sent during the transmission of encrypted packets, to transmit PCR for example, then an extra transport packet may be inserted with an adaptation field but no payload at all, in order to preserve the encryption alignment.

[0068] A method to signal in each frame which 184 bytes block is encrypted, and which one is not is now described. A header made of a few bytes in each DVD sector is used. One byte indicates the number of bytes in the payload before the beginning of the first 184 byte block. Then 11 groups of 2 bits may be used to store the MPEG-2 transport scrambling control field. One bit indicates if the sector contains any encrypted data. A total of 31 bits may be required. Those bits may be stored over an unused DVD sector packet header field, like SCR, when the VOBS has encrypted content. Another option may be to simply encrypt all data, and set a flag in a global header.

[0069] In summary, video elementary stream may be partitioned in blocks of 184 consecutive bytes, and each of these blocks which only contain slice data and only slice data can be encrypted. To restore the alignment of these blocks with the payload of a transport packet, padding bytes may be used in the adaptation field of the transport packet preceding an encrypted transport packet. To preserve the alignment of the payload when an adaptation field is required, an extra packet with no payload may be inserted.

[0070] Allowing the occasional insertion of padding bytes and packets without payload, the bit rate of the video elementary stream may be carefully adjusted to avoid any video buffer overflow. This constraint may be combined with the bandwidth requirements of watermark information.

[0071] Watermarking Support.

[0072] An example of current watermark technology is illustrated in **FIG. 8**. Watermark sectors **800** are performed on this non-HD systems by inserting private sectors **801** at authoring time to store watermark information (mainly replacement data and location for each watermark). The location of a watermark is identified by 3 parameters: A physical sector number **802**, an offset in the sector **803**, and the length in bytes **804**. When a sector **831**, **832**, and **833** is received from data on the disc **830**, the sector number is compared with those in the watermark table and if a replacement is required, replacement data **805** is written into the sector **832** at the location indicated by the offset.

[0073] Refering to **FIGS. 9**, **10A**, and **10B**, we will now discuss the aspect of the present invention that implements

content watermarking. **FIG. 9** is a block diagram of HD running sectors as performed by an exemplary aspect of the present invention where watermark technology may be preserved by only changing location information of the watermarks. **FIG. 10B** is a block diagram of an exemplary aspect of the present invention depicting a watermark marker structure.

[0074] In the HD-DVD context, the receiver is in charge of inserting watermarks and has no knowledge of DVD physical sectors. A solution to this problem is to assign an identifier to each watermark and insert the identifier in the video where the watermark must be inserted as illustrated in **FIG. 10**. In **FIG. 10A**, markers **1002**, **1004**, and **1006** are inserted in the video content **1000** where a watermark is intended to be written. The markers **1050** includes a start code **1052** and a watermark ID **1054**. One skilled in the art will recognize that many different marking schemes may be used to indicate locations for watermark insertions.

[0075] As presently illustrated, the current embodiment uses an 8 byte watermark that is overwritten. The ID **1052** may be a 4 byte long watermark start code such as 0×000001BA. The sequence number **1054** may be a 4 byte unique watermark identifier, WMID **920**. The watermark sector number **802** and offset **803** used in the prior art are replaced by the WMID **920** in the watermark sector. The original 8 bytes of data may be saved in the watermark sector and the PTS of the picture to which the watermark applies allows the transport stream multiplexer **214** to send the watermark data in real-time. For example, the WMID **920** may be an incrementing counter starting at 0×0200 to avoid generating start codes.

[0076] An alternative method would to use the WMID/offset **920** as an offset into the frame. This method would not require any markers in the video data.

[0077] As described above, the watermarks are stored in HD watermark sectors **900**. A group of watermarks with the same PTS **922** may be referred as a watermark access unit and may be stored in the same physical sector. This access unit may follow a watermark buffering model, which may be described with a leak rate and buffer size that may be defined depending on the bandwidth allowed for watermarks (This buffering model is described the MPEG-2 standard). The transport multiplexer **214** may ensure that each access unit arrives in time in the watermark buffer.

[0078] When a picture is received and watermarks have to be inserted, the picture may be scanned for the watermark start codes which are followed by the WMID **920**. The watermark buffer **910** has the corresponding WMID **920** information to either restore the original 8 bytes **924** (start code and WMID) or to insert the replacement data **928**. The size of the replacement data **926** may be stored as part of the watermark buffer **910**. If the corresponding WMID **920** is not in the buffer, then a pirate attack is very likely to have occurred. The TV may decide to wait a few seconds and turn the screen dark, refusing content playback.

[0079] If the start code search method is too demanding in CPU resources in the TV, an offset from the first byte of the slice data could indicate the location of each watermark.

[0080] Encryption of watermarks may be done on a watermark access unit level. Watermarks belonging to the same frame (i.e. watermarks having the same PTS) may be grouped together in a more efficient manner to allow encryption: a watermark access unit header followed by watermark data. The header could be composed of the DTS, number of watermarks in the access unit, size in bytes, and would not be encrypted. The rest of the data could be encrypted and aligned by the transport multiplexer **214** with the same method that for video access units. Not encrypting the header should not compromise the security of the system since the WMID found in the picture at watermark insertion time must match the watermark data, and watermarks attacks can be detected.

[0081] **FIG. 11** is a block diagram of an exemplary aspect of the present invention depicting a content processor **1110** that is configured to input elementary streams, insert watermarks, perform trick mode display functions, multiplex audio and video content, and formats the resultant data into a valid output transport stream **1134** such as ATSC for output. The elementary streams include watermarking packets **1100**, video packets **1102**, and audio packets **1104**. The watermarking packets are input into a watermarking buffer **1120**. A watermark inserter **1126** inputs watermark sectors from the watermark buffer **1120** and inserts watermarks into the video data stored in a first video buffer **1122** for output into a second video buffer **1128** using watermarking techniques that were discussed previously. The video packets **1102** are input into a first video buffer **1122**. The data is then transferred into a second video buffer **1128** where the video data is combined with watermarks. Next, the data stored in the second video buffer **1128** may be input to a trick mode processor **1130** where output display trick modes may be performed on the video streams. The audio packets **1104** are input into an audio buffer **1124**. Data processed by the trick mode processor **1130** and the audio buffer **1124** are both input into a content multiplexer **1132** which combines the data into a combined data stream. The combined data stream is then input into a transport packet generator **1134**, which formats the data into a transport packet stream **1140** such as ATSC. One skilled in the art will recognize that a content processor could be built to handle other types of data instead of or in combination with the watermark, video and audio data types discussed here.

[0082] **FIG. 15** shows a block diagram of another embodiment of the present invention demonstrating how video watermark insertion may occur in an HDTV **1520**. As illustrated, the content is contained on a media **1500**. The content is read and processed by an HD player **1510** which produces a transport stream such as ATSC and modulated using a modulation scheme such as 8-VSB, containing processed content for display. The processed content is input into the HDTV, where it may be decrypted and demultiplexed by a decrypter/demultiplexer device **1530**. Next the data is stored into buffers. In this example, the timestamped watermark elementary stream is buffered in a timestamped watermark elementary stream buffer **1532**. The timestamped video elementary stream is buffered in a timestamped video elementary stream buffer **1534**. Both the buffered watermark and video data are input into a watermark inserter **1538** where watermarks are inserted into the video stream producing a watermarked video stream **1540** such as MPEG video. The watermarks are inserted when the timestamps (DTS/ PTS) match the current Video picture DTS/PTS time stamps The watermarked video stream **1540** is then displayed as a decoded image **1542**.

[0083] One skilled in the art will appreciate that the concept of watermarking as presented may equally be applied to other types of data streams besides video, such as audio, executable or process data. Executable data may include programs intended for execution on a target device such as a smart HDTV. Process data may include data or files that communicate information such as HTML'or XML to a target device.

[0084] Trick Modes

[0085] Trick modes modify the video stream to produce output display effects such as pause, slow motion, fast forward and reverse. Traditionally, trick modes are generated directly by decompression chips. The present invention may generate trick modes altering the video stream before it is decoded. **FIG. 12** shows a video stream **1200** being input to a trick mode processor **1210**. The output of the trick mode processor **1210** is a modified video stream **1220** that may now be multiplexed with other content streams before being converted into a stream of transport packets. The video frames are typically MPEG frames. MPEG frames include P frames, B frames and I frames. I frames, are video frames known as Intra-coded pictures (I-pictures). I frames are coded in such a way that they can be decoded without knowing anything about other pictures in a video sequence. P frames, are video frames known as predictive coded pictures (P-pictures). P frames are decoded using information from another frame that was displayed earlier. B frames, are video frames known as bidirectionally predicted pictures (B-pictures). B frames are bi-directionally decoded using information from other frames. The other frames may occur before or after the B frame. P frames and B frames are often referred to as predictive frames. Trick modes may be achieved by extracting MPEG-2 video elementary frames using search algorithms. The frames may be converted to a valid MPEG-2 video elementary stream by adjusting headers, like the temporal reference fields of picture headers and by inserting empty P frames or empty B frames. An empty frame has null motion vectors, no residual data coded (coded block pattern is 0) and has the property of repeating the content of one of the reference frames. These techniques, along with a time stamp correction provides the possibility to generate a valid MPEG-2 elementary video stream with a valid number of frames per second (29.97 for NTSC for example). The impact on the content processor **214** is that it must continuously output the data stream. A stack of queued transport packets transferred in hardware by DMA may reduce the amount of CPU required.

[0086] Each picture header and PES (program elementary stream) header may be changed to reflect the insertion or deletion of pictures in the elementary video stream. Because of the interdependency of I, P and B frames, some rules may need to be followed including: (1) any B frame may be suppressed or inserted; (2) a P frame may be suppressed only if all other frames dependent upon the suppressed P frame are also suppressed.

[0087] **FIG. 13** shows a sequence of frames where the first frame **1300** is a first original picture. Empty B frame(s) **1310** may be inserted into the video stream to create a slow motion or pause effect. Then a second original picture **1320** is input to the video stream. Fast forward and rewind playback may be obtained by playing back I frames and inserting empty B frames to adjust playback speed and control the bitrate.

[0088] Although the trick modes are described here in terms of MPEG-2 frames, one skilled in the art will recognize that the present invention may be practiced on other types of video that utilize predictive video frames.

[0089] Data Flow

[0090] **FIG. 14** is a block diagram that shows the data flow through an embodiment of the present invention during playback of high definition content with real-time conversion to a packet transport stream. The high definition digital content is authored and stored on storage media **1400**. An HD player **1410** may then read the content from the media **1400**. The content may be stored as audio, video, and data sectors **1420**. An example of a storage media may be a classic DVD disc, extended with High Definition Video formats and an example of a data sector type may be DVD sector. A media player **1430** may include a media reader and media reader logic. Data may be extracted from the sectors **1430** and demultiplexed into elementary streams including an elementary video stream **1440**, an elementary watermark stream **1442**, and an elementary audio stream **1444**. Timestamps may be included within the elementary data streams. These streams may be input to a content processor **1450** where they may be processed. Processes may include insertion of watermarks, processing trick output display modes, multiplexing content streams, and transport packet generation. The output of the content processor **1450** may be transport packets such as ATSC transport packets. Content multiplexing may need to follow packet alignment methods to ensure valid decrypted elementary streams when the streams are encrypted. A modulator **1460** may modulate the output packets for transport to an HDTV **1470**. The HDTV may also perform functions on the content including demultiplexing the elementary streams, decoding the content, decrypting the content, watermarking the content and displaying the content.

[0091] The present invention provides extensions of media formats including DVD to high resolution video, while maintaining most of the current architectures. An added benefit of this invention is backward compatibility, although backward compatibility may be limited to some audio and video format. These HD-DVD extensions provide for encrypting content, watermarking content, and trick playback display modes.

[0092] Although the present invention has been fully described by way of examples with reference to the accompanying drawings, it is to be noted that various changes and modifications will be apparent to those skilled in the art. For example, it will be apparent to those of skill in the art that the content may be provided from any type of source device for processing and playback on other devices according to principles of the present invention. Therefore, unless such changes and modifications depart from the scope of the present invention, they should be construed as being included therein.

**1-24**. (canceled)

**25**. An apparatus for playing high definition content comprising:

(a) a high definition media player for receiving multiplexed high definition content from a media source, said multiplexed high definition content and timestamps contained in data packets; and

(b) a content processor for processing said multiplexed high definition content into transport packets.

**26**. The apparatus according to claim 25, wherein said data packets are contained in sectors.

27. The apparatus according to claim 25, further including a transport packet modulator for modulating said transport packets.

28. The apparatus according to claim 25, wherein said data packets comprise at least one of:

(a) watermark data;

(b) video data;

(c) audio data;

(d) executable data; and

(e) process data.

29. The apparatus according to claim 25, wherein preselected bocks of said data packets are encrypted.

30. The apparatus according to claim 25, wherein said media source is an optical disc.

31. The apparatus according to claim 26, wherein said sectors are DVD sectors, and said content processor generates an ATSC transport stream.

32. The apparatus according to claim 28, wherein said watermark data includes at least one of:

(a) a watermark identifier;

(b) an offset;

(c) a presentation time stamp;

(d) original data; and

(e) size data.

33. The apparatus according to claim 28, wherein said video data includes at least one of the following:

(a) a start code; and

(b) a watermark identifier.

34. The apparatus according to claim 25, wherein said content processor further includes a trick mode generator.

35. The apparatus according to claim 34, wherein said trick mode generator can create a slow motion effect by inserting empty predictive frames into a video elementary stream between picture frames, wherein the rate of said slow motion effect is determined by the quantity of predictive frames inserted.

36. The apparatus according to claim 34, wherein said trick mode processor can create a pause effect by inserting into a video elementary stream a multitude of predictive frames.

37. The apparatus according to claim 34, wherein said trick mode processor can create a fast forward playback effect by inserting forwardly sequenced Intra-coded picture frames interspersed with empty predictive frames into a transport packet stream, wherein the rate of said fast forward motion effect is determined by the quantity of Intra-coded picture frames and predictive frames.

38. The apparatus according to claim 34, wherein said trick mode processor can create a rewind playback effect by inserting reverse sequenced Intra-coded picture frames interspersed with empty predictive frames into a transport packet stream, wherein the rate of said rewind playback effect is determined by the quantity of Intra-coded picture frames and predictive frames.

39. The apparatus according to claim 34, wherein said trick mode processor can create a playback effect by inserting Intra-coded picture frames interspersed with frames into a transport packet stream.

40. The apparatus according to claim 25, further including an HD-TV comprising:

(a) a decrypter;

(b) a demultiplexer;

(c) a watermark buffer for receiving watermark data;

(d) a video buffer for receiving video data; and

(e) a watermark inserter for inserting watermarks into the video data, determined by the video data and watermark data.

41. A method for playing multiplexed high definition content comprising:

(a) receiving multiplexed high definition content from a media source, said multiplexed high definition content and timestamps contained in data packets; and

(b) processing said multiplexed high definition content into transport packets.

42. The method according to claim 41, wherein said data packets are contained in sectors.

43. The method according to claim 41, further including the steps of:

(a) modulating said transport packets; and

(b) outputting said modulated transport packets.

44. The method according to claim 41, wherein said step of receiving said multiplexed high definition content from a media source comprises reading content from an optical disc.

45. The method according to claim 44, wherein said optical disc is a DVD and said step of reading content from an optical disc further comprises reading DVD sectors from said optical disc.

46. The method according to claim 41, wherein said step of processing said multiplexed high definition content into transport packets further includes generating a slow motion effect by inserting empty predictive frames into a video elementary stream between picture frames, wherein the rate of said slow motion effect is determined by the quantity of predictive frames inserted.

47. The method according to claim 41, wherein said step of processing said multiplexed high definition content into transport packets further includes generating a fast forward playback effect by inserting forwardly sequenced Intra-coded picture frames interspersed with empty predictive frames into a transport packet stream, wherein the rate of said fast forward motion effect is determined by the quantity of Intra-coded picture frames and predictive frames.

48. The method according to claim 41, wherein said step of processing said multiplexed high definition content into transport packets further includes generating a rewind playback effect by inserting reverse sequenced Intra-coded picture frames interspersed with empty predictive frames into a transport packet stream, wherein the rate of said fast forward motion effect is determined by the quantity of Intra-coded picture frames and predictive frames.

49. The method according to claim 41, wherein said step of processing said multiplexed high definition content into transport packets further includes creating a playback effect by inserting Intra-coded picture frames interspersed with frames into a transport packet stream.

* * * * *

# A SINGLE-CHIP MPEG-2 MP@ML AUDIO/VIDEO ENCODER/DECODER WITH A PROGRAMMABLE VIDEO INTERFACE UNIT

*C.T. Chen, T.C. Chen, F-C. Jeng, H. Cheng, K. Konstantinides*

Stream Machine
San Jose, CA
USA

## 1. ABSTRACT

We present a single-chip, MPEG-2 Main Profile at Main Level, audio and video encoder and decoder. It combines a RISC core, a 24-bit DSP, video and audio interface units, and several dedicated processing units. A programmable video interface unit supports multiple modes of pre- and post-processing and on-screen display (OSD). The codec has been implemented using a standard-cell library in 0.18 $\mu m$ CMOS technology.

## 2. INTRODUCTION

The adoption of the MPEG standard [1] offered consumers a new generation of products, such as DVD players, digital TV, and personal video recorders. While the first generation of MPEG-based products offered playback-only capability, the latest cost effective MPEG-encoding solutions[2, 3] allow for a new class of affordable digital video recording products. These codecs integrate complete MPEG-2 video encoders and decoders; however, a complete digital audio/video system still requires additional hardware for audio encoding and decoding and for multiplexing or demultiplexing the audio and video streams.

In this paper we present an MPEG-2 ML@MP codec that pushes system integration even further, by integrating both audio and video encoding and decoding into a single chip. In addition to real-time audio and video coding, this codec provides programmable support for multiplexing and demultiplexing, pre- and post-processing of video data, and on-screen display (OSD). These combined benefits make this codec an ideal single-chip solution for a variety of MPEG-2-based applications, such as SVCD recorders or USB-based TV/video players and recorders.

## 3. SYSTEM ARCHITECTURE

Figure 1 shows the major functional units of the MPEG A/V codec. These units include: the RISC microcontroller, the Video Interface Unit (VIO), the Audio Interface Unit (AIU), the Video Engine Unit (VEU), the Audio Engine (DSP), the Host Interface Unit (HIU), and the SDRAM Control Unit (DCU).

All blocks inter-communicate using two major buses: a 64-bit wide data bus (D-Bus) and a 16-bit wide register bus (R-Bus). In addition to the above seven major blocks, the $I^2C$ CTRL block provides control for external NTSC/PAL video encoders and decoders. The PLL block provides clocking for all internal blocks and also external memory. Given an input 27 MHz clock, all internal components operate at 108 MHz. A separate audio PLL is used to provide an output clock for external audio A/D and D/A converters.

### 3.1. The RISC Microcontroller

This is an embedded, programmable, 32-bit ARC RISC processor [4]. It performs multiplexing of audio and video elementary streams and demultiplexing of MPEG program streams. It also acts as a central controller and sequencer. Its microcode can be downloaded either from an external host or from an external EPROM or Flash memory, through the Host Interface Unit.

The embedded software design effort for such a codec requires code development for two distinct type of tasks: timing-critical tasks, such as video compression, and non-timing-critical tasks, such as multiplexing of audio and video and user communications. One solution for such a system is to use a single RISC processor running a real operating system. In this case context switching time is very important and unless the RISC processor is very powerful it is very difficult to guarantee a predictable behavior for timing critical tasks.

Our solution is different. The RISC core features novel memory mapping and interrupt controlling schemes that allows it to handle both critical timing requirements and traditional software applications, without the need to run a real-time operating system. Specifically, we have dedicated interrupt vectors, and memory (data and instruction) to two

0-7803-7041-4/01/$10.00 ©2001 IEEE

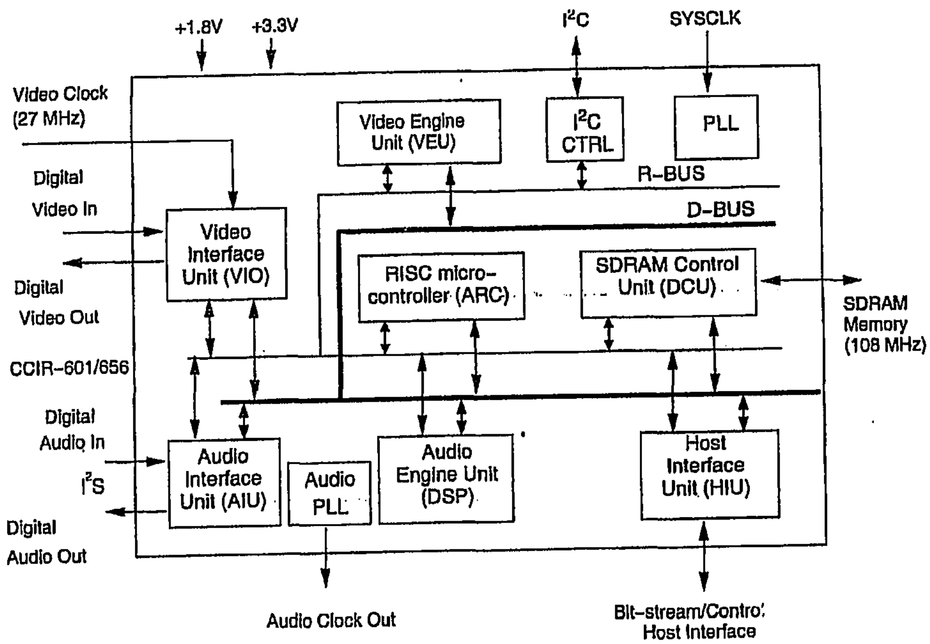



**Fig. 1.** Block diagram of the MPEG-2 audio/video codec.

different types of tasks: time-critical and non-critical. Since all time-critical tasks are interrupt driven and have their own memory space, there is no need for context switching. This allows for easier software development and predictable performance.

### 3.2. The Host Interface Unit

The host interface is used to communicate with the host controller and external EPROM or flash memory. It supports a variety of communication protocols, including 16-bit Motorola- or Intel-like interfaces, and a generic 8-bit interface. The host interface has a glue-less interface to USB controllers and it may also be used in PC-based host systems using a PCI bridge interface. The HIU is also used for the I/O of the compressed bit streams between the codec and an external controller.

### 3.3. The Audio Interface Unit (AIU)

The audio interface unit provides the interface between the codec and external audio devices. Audio samples are transferred in and out of the codec using $I^2S$ signaling. The codec also provides a user-configurable output clock for external audio A/D and D/As.

### 3.4. The Video Engine Unit (VEU)

Figure 2 shows a block diagram of the VEU. It includes a

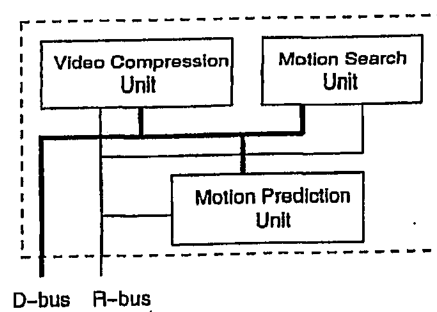



**Fig. 2.** Block diagram of Video Engine Unit.

video compression unit (VCU), a motion search unit (MSU), and the motion prediction unit (MPU). The VEU is the video processor core for the codec. During encoding, it operates on the video data and generates an MPEG-compliant video elementary stream. Among its many functions, it performs motion estimation and compensation, DCT, quantization, rate control, and variable length coding.

During decoding, it operates on a video elementary stream and generates decompressed video frames. It performs vari-

When the VIO is configured in the advanced mode, input video can be mixed directly with OSD data and then passed back to the VIU and then to SDRAM for video encoding. Applications of this mode include the initial encoding screen menu set up, and real-time video scaling and editing at encoding. Using this advanced mode, users can also blend text and graphics into the input video that is being encoded.

### 4.2. Video Decoding Mode

Fig. 5 shows the flow of data in the VIO during video decoding. At minimum processing, decoded video data are

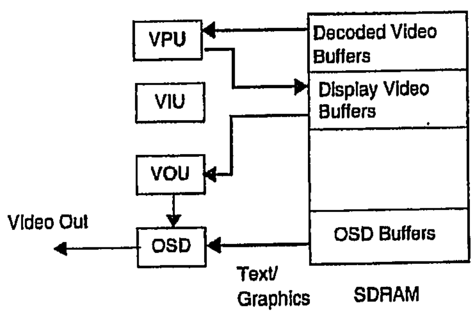



**Fig. 5.** VIO - Decoding mode.

transferred from the SDRAM to the VOU for chroma upconversion and other postprocessing. The output of the VOU is passed to the OSD where it can be mixed with text or graphics before it is transferred to the video output. Optionally, the VPU may also be enabled to process the decoded data before they are being transferred to the VOU. For example, the VPU can be used to scale-down specific video frames to create a thumbnail screen. Table 1 summarizes the key features of the codec.

| | |
|---|---|
| Video coding | MPEG-1, MPEG-2 MP@ML, SP@ML, I/P/B frames |
| Audio coding<br><br>Resolutions | Dolby Digital, MPEG (all layers)<br>up to 24-bit/sample |
| Resolutions<br><br>Rates | D1, 2/3 D1, 1/2 D1<br>CIF, SIF, QCIF<br>24, 25, 29.97, 30 Hz |
| Prediction<br><br><br>Rate Control | Adaptive field/frame/16×8<br>Adaptive frame/field DCT<br>Adaptive inter/intra<br>CBR, VBR (one-pass)<br>I-only to 30 Mb/s |
| Interfaces | Video: ITU-601/656<br>$I^2C$,$I^2S$,SDRAM,EPROM,<br>Flash,16-b/8-b parallel |
| Other | pre-, post-filtering<br>4:2:2 to 4:2:0<br>4:2:0 to 4:2:2<br>temporal/spatial filtering<br>telecine, inverse telecine<br>8-bit OSD |

**Table 1.** Key features of the Audio/Video Codec

## 5. IMPLEMENTATION AND STATUS

The codec is implemented using a standard-cell library in 0.18 $\mu m$ CMOS technology. It uses an 108 MHz system clock.

## 6. CONCLUSIONS

In this paper we presented the architecture of single-chip MPEG-2, MP@ML, audio/video codec. By taking into consideration the overall system requirements in consumer-based digital video recording, we designed the codec with a unique and flexible video interface unit. The VIO can accommodate a variety of video pre- and postprocessing algorithms, thumbnail processing/editing, and loopback, in a very efficient way. Used with a standard DVD decoder, the codec can provide full-duplex DVD playback and recording functionality for time-shift or DVD-recordable applications.

## 7. ACKNOWLEDGMENTS

We would like to thank all the hardware and firmware engineers that have contributed to the design and implementation of this codec.

## 8. REFERENCES

[1] ISO/IEC JTC1 CD 13818. Generic coding of moving pictures and associated audio. International Organization for Standardization (ISO), 1994.

[2] C.T. Chen et al. A single-chip MPEG-2 video encoder/decoder for consumer applications. In *Proc. ICIP-99*. IEEE, 1999.

[3] C-Cube Microsystems. Dvxpress-mx25 and dvxpress-mx50. Product Brief, www.c-cube.com, 1999.

[4] Argonaut Technologies, Ltd. *ARC Interface Manual*, 1999.